Final Report • March 1995

HYDROLIGHT 3.0 USERS' GUIDE

Submitted by:

Dr. Curtis D. Mobley
Applied Electromagnetics and Optics Laboratory

Prepared for:

Dr. Steven Ackleson
Environmental Optics Program, Code 3220P
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

SRI Project 5632
Contract N00014-94-C-0062

Approved:

Joseph E. Nanevicz, Director
Applied Electromagnetics and Optics Laboratory

Murray J. Baron, Vice-President
Advanced Development Division
Engineering Research Group

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1995	Final Report

4. TITLE AND SUBTITLE

HYDROLIGHT 3.0 USER'S GUIDE

5. FUNDING NUMBERS

N00014-94-C-0062

6. AUTHOR(S)

Dr. Curtis D. Mobley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

8. PERFORMING ORGANIZATION REPORT NUMBER

SRI Project 5632

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Steven Ackleson
Environmental Optics Program, Code 3220P
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

HYDROLIGHT 3.0 is a radiative transfer numerical model that computes spectral radiance distributions and related quntities for natural water bodies. The model solves the time-independent, one-dimensional radiative transfer equation using computationally efficient techniques based on invariant imbedding theory. Input to the model consists of information about the absorbing and scattering properties of the water body, about the sky radiance incident on the water surface, about the state of the wind-blown sea surface, and about the bottom of the water column. The water optical properties can vary arbitrarily with depth and wavelength. The model includes internal sources such as bioluminescence, fluorescence by chlorophyll and by dissolved organic matter, and Raman scattering. Output from the model includes the spectral radiance as a function of depth, direction, and wavelength as well as derived quantities such as irradiances, reflectances, and diffuse attenuation functions.

This report is a self-contained guide to running the HYDROLIGHT 3.0 model, which is written in FORTRAN 77. The report first gives a descriptive overview of the HYDROLIGHT 3.0 model. Input to the model is then described in detail. The report includes graphical examples of the model's output.

14. SUBJECT TERMS

Optical oceanography, radiative transfer, remote sensing, numerical modeling

15. NUMBER OF PAGES

65

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	Unlimited

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

General knowledge of the radiance distribution within and leaving a water body is a prerequisite for the solution of many problems in underwater visibility, remote sensing, biological primary productivity, and mixed-layer thermodynamics. Moreover, because radiance is the fundamental radiometric quantity, all other quantities of interest to optical oceanographers, such as the various irradiances, K-functions, and reflectances, can easily be computed once the radiance is known.

HYDROLIGHT 3.0 is a radiative transfer numerical model that computes radiance distributions and derived quantities for natural water bodies. In brief this model computes from first principles the time-independent radiance distribution within and leaving any plane-parallel water body. Input to the model consists of the absorbing and scattering properties of the water body, the nature of the wind-blown sea surface and of the bottom of the water column, and the sun and sky radiance incident on the sea surface. Output consists both of archival printout and of files of digital data, from which graphical or other analyses can be performed.

Note that the HYDROLIGHT model *per se* is a radiative transfer model, not a model of oceanic optical properties. The user must therefore supply the water inherent optical properties (IOP's) to the HYDROLIGHT core code. However, this means that HYDROLIGHT is not restricted in application to any particular water body or optical type of water (or indeed, even to water bodies in general, although that is our interest here). HYDROLIGHT does contain, however, a number of sub-models of water IOP's, sky conditions, and bottom types. These sub-models can be used as provided to solve a wide range of problems in optical oceanography and limnology, or they can serve as templates for user-written sub-models for the various kinds of required input.

The input IOP's, namely the absorbing and scattering properties of the water body, can vary arbitrarily with depth and wavelength. These IOP's can be obtained from actual measurements or from analytical models, which can build up the IOP's from contributions by any number of components. The input sky radiance distribution can be completely arbitrary in the directional and wavelength distribution of the solar and diffuse sky light. In its most general solution mode, HYDROLIGHT 3.0 includes the effects of inelastic scatter by chlorophyll fluorescence, by colored dissolved organic matter (CDOM) fluorescence, and by Raman scattering by the water itself. The model also can simulate internal layers of bioluminescing microorganisms

This report is a users' guide for running version 3.0 of the HYDROLIGHT model, which was released in March 1995. It is assumed that the reader is familiar with the basic terminology and notation of optical oceanography. [If this is not the case, then the reader should consult the overview by Mobley (1995) or one of the books by Kirk (1994), Spinrad, Carder, and Perry (1994), or Mobley (1994).] This report gives a general overview of the capabilities of HYDROLIGHT 3.0, describes in detail the input required to run the model, and shows example output. This report is independent of any other publication and should be adequate for users who wish to run HYDROLIGHT 3.0 as a "black box" model. However, the report freely references various publications, so that interested users can locate additional documentation. The text *Light and Water: Radiative Transfer in Natural Waters* (Mobley, 1994) describes in considerable detail the mathematical methods employed in HYDROLIGHT 3.0. That book is the primary technical documentation for the HYDROLIGHT 3.0 model, and the source code itself contains extensive comments that refer to *Light and Water*.

Throughout this report, the names of mathematical variables are written in italics, for example, *U*, *nray*, or *zeta*. This report and the HYDROLIGHT code follow the standard FORTRAN convention that variables beginning with the letters *i-n* are integers, and variables beginning with *a-h* and *o-z* are floating-point numbers. The names of computer programs and of computer files are written in a sans serif font, for example, inishl1, abcase1.f, or Output2.ex1.

2. OVERVIEW OF HYDROLIGHT 3.0

Many problems of interest in oceanography and remote sensing can be solved using time-independent radiative transfer theory applied to plane-parallel geometries. The consideration of time-dependent, plane-parallel problems is not as restrictive as it might seem on first glance. For example, although the oceans are horizontally inhomogeneous, the horizontal scales of variability (typically tens of meters to kilometers) are usually much greater than the vertical scales (tens of centimeters to tens of meters). In this case we can think of the ocean as consisting of optically independent "patches" of water, for which each patch can be well modeled as a horizontally homogeneous water body whose optical properties vary only with depth. (This is a one-dimensional geometry, with the one dimension being depth.) We can then independently apply a one-dimensional radiative

transfer model at the center of each patch in order to simulate the entire, horizontally inhomogeneous water body. In the analysis of imaging spectrometer data, one can even apply such a model to the water patch associated with each pixel in the image, although such high spatial resolution often is unnecessary.

Such a piecewise simulation is justified so long as the horizontal size of each water patch is at least several photon mean free paths. This is usually the case. In the open ocean, photon mean free paths (the inverse of the beam attenuation coefficient) are never more than 50 m (at blue wavelengths, and much less at other wavelengths) in even the clearest waters; horizontal variability in such waters is often on scales of kilometers. In coastal waters subject to river runoff, sediment resuspension, and variable shallow bottom topography, optical properties and boundary conditions can change horizontally on scales of meters to tens of meters. However, such waters tend to be rather turbid, with photon mean free paths of tens of centimeters to a few meters. In either case, the use of a one-dimensional radiative transfer model is justified.

The use of time-independent radiative transfer is valid whenever the time scales for changes in environmental conditions (typically seconds to seasons) are much greater than the time required for the light field to assume a steady state within the water body after a change in the optical properties or boundary conditions (milliseconds). Solving a sequence of time-independent, one-dimension radiative transfer problems in order to simulate a changing (in both time and space) water body is computationally much faster than solving one large time-dependent, three-dimensional problem.

2.1 General Description

The fundamental quantity that describes the time-independent, one-dimensional light field in the ocean is the spectral radiance $L(z,\theta,\phi,\lambda)$, which has units of W m^{-2} sr^{-1} nm^{-1}. The spectral radiance fully determines the depth (z), directional (θ,ϕ), and wavelength (λ) behavior of the light field. Therefore, all other quantities of interest, such as various irradiances, K-functions, and reflectances, can be computed from their definitions once the spectral radiance is known. In order to predict the spectral radiance, HYDROLIGHT solves the integro-differential radiative transfer equation (RTE) along with its boundary conditions. Because of their mathematical complexity, these equations must be solved numerically for any realistic situation.

In the HYDROLIGHT model, the depth z, in meters, is measured as positive downward from zero at the mean sea surface; θ is the polar angle, measured from zero in

the nadir (downward) direction; ϕ is the azimuthal angle, measured from zero in the downwind direction; and wavelength λ is measured in nanometers. The direction (θ,ϕ) refers to the direction that the photons generating the radiance are traveling, which is opposite to the viewing direction. Thus, for example, the zenith (or nadir-viewing) radiance refers to light heading upward, which is detected by an instrument pointed downward. [This convention on directions is standard in radiative transfer theory. Observational oceanographers usually find it more convenient to let *their* (θ,ϕ) specify the viewing direction, i.e., the direction that their instrument is pointed.]

Any radiance sensor actually measures an average of $L(z,\theta,\phi,\lambda)$ taken over some finite solid angle $\Delta\Omega$, which is determined by the field of view of the instrument, and over some finite bandwidth $\Delta\lambda$, which is determined by the wavelength response of the instrument. Likewise, in order to solve the RTE numerically, we discretize it by averaging over direction and wavelength. In the HYDROLIGHT model, this directional averaging is performed by first partitioning the set of all directions (θ,ϕ), $0 \le \theta \le \pi$, $0 \le \phi < 2\pi$, into regions bounded by lines of constant θ and constant ϕ, plus two polar caps. These quadrilateral regions and polar caps are collectively called "quads." The individual quads Q_{uv} are labeled by discrete indices $u = 1, 2, ..., M$ and $v = 1, 2, ..., N$ to show their θ and ϕ positions, respectively. Examples of such quad partitionings are shown later in Fig. 1 on page 16. The lines of constant θ can be arbitrarily spaced. Similarly, the wavelength region of interest, typically 350 to 700 nm, is partitioned into a number of contiguous wavelength bands of width $\Delta\lambda_j$, $j = 1, 2, ..., J$. The $\Delta\lambda_j$ need not be the same size for different j values. The fundamental quantities computed by the HYDROLIGHT 3.0 model are then the quad- and band-averaged radiances at any selected set of depths z_k, $k = 1, 2, ..., K$:

$$L(k,u,v,j) \equiv \frac{1}{\Delta\lambda_j \, \Delta\Omega_{uv}} \int_{\Delta\lambda_j} \int_{Q_{uv}} L(z_k,\theta,\phi,\lambda) \sin\theta \, d\theta \, d\phi \, d\lambda \, . \tag{1}$$

In addition to the radiances within the water, HYDROLIGHT computes the upwelling radiance in all directions (all quads) just *above* the sea surface. This upwelling radiance includes both the water-leaving radiance and that part of the incident direct and diffuse sky radiance that is reflected upward by the wind-blown sea surface. The water-leaving and reflected-sky radiances are computed separately in order to isolate the water-leaving radiance, which is the quantity of interest in most remote sensing applications. The development of the quad- and band-averaged versions of the RTE and of the associated boundary conditions is given in full in Mobley (1994); see Section 8.2 in particular.

In order to run HYDROLIGHT 3.0 to predict the spectral radiance distribution within and leaving a particular body of water, during particular environmental (sky and surface wave) conditions, the user supplies the core model with the following information (via direct input or via user-written subroutines):

- The inherent optical properties of the water body. These optical properties are the absorption and scattering coefficients and the scattering phase function (or their equivalent, such as the volume scattering function, the beam attenuation coefficient, and the albedo of single scattering). These properties must be specified as functions of depth and wavelength.

- The state of the wind-blown sea surface. Version 3.0 of HYDROLIGHT models the sea surface using the Cox-Munk capillary wave slope statistics, which adequately describe the optical reflection and transmission properties of the sea surface for moderate wind speeds and for solar angles and lines of sight away from the horizon. In this case, only the wind speed needs to be specified.

- The sky spectral radiance distribution. This radiance distribution (including background sky, clouds, and the sun) can be obtained from semi-empirical models that are built into HYDROLIGHT, from observation, or from a separate user-supplied atmospheric radiative transfer model.

- The nature of the bottom boundary. For finite-depth bottoms, this information is given in terms of the irradiance reflectance of the bottom. For infinitely deep water, the inherent optical properties of the water body below the region of interest are given, from which the model computes the needed bi-directional radiance reflectance.

The absorption and scattering properties of the water body can be provided to the HYDROLIGHT model in various ways. For example, if actual measurements of total absorption are available at selected depths z and wavelengths λ, then these values can be read from a file provided at run time. An interpolation scheme can be used to define absorption values for those z and λ values not contained in the data set. In the absence of actual measurements, the total absorption of the water body can be modeled in terms of contributions by any number of components. Thus the total absorption can be built up as

the absorption by water itself, plus the absorption by chlorophyll-bearing microbial particles, plus that by CDOM, by detritus, by mineral particles, and so on. In order to specify the absorption by chlorophyll-bearing particles, for example, the user can specify the chlorophyll profile of the water column and then use a bio-optical model to convert the chlorophyll concentration to the needed absorption coefficient. The chlorophyll profile also provides information needed for the computation of chlorophyll fluorescence effects. Each such absorption component has its own depth and wavelength dependence. Similar modeling can be used for scattering.

Phase function information is generally provided by using a Rayleigh-like phase function for scattering by the water itself, by using a Petzold-like phase function for scattering by particles, and by assuming that dissolved substances like CDOM do not scatter. The individual-component phase functions are weighted by the respective scattering coefficients and summed in order to obtain the total phase function.

HYDROLIGHT also requires the downwelling radiance incident onto the sea surface as input. The HYDROLIGHT model does not carry out any radiative transfer calculations for the atmosphere *per se*. The sky radiance for either cloud-free or overcast skies can be obtained from simple analytical models or from a combination of semi-empirical models. Such models are provided as a part of the HYDROLIGHT code. Alternatively, if the sky radiance is measured, that data can be used as input to HYDROLIGHT via a user-written subroutine. It is also possible to run an independent atmospheric radiative transfer model such as LOWTRAN (Kneizys et al., 1988) in order to generate the sky radiance coming from each quad of the sky hemisphere, and then give the model-generated values to HYDROLIGHT as input.

The bottom boundary condition is applied at the deepest depth *of interest* in the simulation at hand. Consider first the case of water that is optically infinitely deep. For a remote sensing simulation concerned only with the water-leaving radiance, it may be sufficient to solve the radiative transfer equation only for the upper two "diffuse attenuation depths," because almost all light leaving the water surface comes from this near-surface region. [The diffuse attenuation depth $1/K_d$ can be roughly estimated from simple bio-optical models such as Eq. (3.50) of *Light and Water*.] In this case, the bottom boundary condition can be taken to describe an infinitely deep layer of water below the corresponding geometric depth. In a biological study of primary productivity, it might be necessary to solve for the radiance down to five (or more) diffuse attenuation depths, in which case the bottom boundary condition would be applied at that geometric depth. In such cases HYDROLIGHT computes the needed bottom boundary information from the

inherent optical properties at the deepest depth of interest. The bottom boundary condition also can describe a physical bottom at a given geometric depth. In that case, the Lambertian irradiance reflectance of the bottom must be specified. In general, this reflectance is a function of wavelength and depends on the type of bottom – mud, sand, brown algae, green algae, and so on.

Output from HYDROLIGHT consists of both "printout" (an ASCII file suitable for hardcopy printing) and a file of binary digital data. The default printout gives a moderate amount of information to document the input to the run and to show selected results of interest to most oceanographers (such as various irradiances, reflectances, mean cosines, irradiance K-functions, and zenith and nadir radiances). Optionally, the printout can give the full radiance distribution (separated into direct and diffuse components), radiance K-functions, path functions, and the like. The file of digital data contains the complete output from the run. This file is generally used as input to plotting routines that give graphical output of various quantities as functions of depth, direction, or wavelength. Routines for graphical output are not a part of the HYDROLIGHT code because of the wide variety of graphics packages in use and because different users generally want different kinds of plots. However, several plotting routines written in the IDL language (see IDL in the references) are included with the HYDROLIGHT code as a convenience for users who have that popular software package.

2.2 Ways in Which HYDROLIGHT 3.0 Can Be Used

Previous versions of HYDROLIGHT have been used in a variety of studies ranging from bio-optical oceanography to remote sensing. Some of the ways in which HYDROLIGHT can be used are as follows:

- HYDROLIGHT can be run with modeled input values to generate in-water light fields, which in turn become the input to models of primary productivity or mixed-layer dynamics. Such information is fundamental to the coupling of physical, biological, and optical feedback models.

- HYDROLIGHT can be run with the IOP's of different water types to simulate in-water light fields for the purpose of selecting or designing optical instruments for use in various water types. Such information can aid in the planning of field experiments.

- HYDROLIGHT can be run with assumed water IOP's as input, in order to obtain estimates of the signals that would be received by various types or configurations of remote sensors, when flown over different water bodies and under different environmental conditions. Such information can guide the planning of specific operations.

- HYDROLIGHT can be used to isolate and remove unwanted contributions to remotely sensed signatures. Consider the common remote-sensing problem of extracting information about a water body from a downward-looking imaging spectrometer. The detected radiance contains both the water-leaving radiance (the signal, which contains information about the water body itself) and sky radiance reflected upward by the sea surface (the noise). The HYDROLIGHT model separately computes each of these contributions to the radiance heading upward from the sea surface and thus provides the information necessary to correct the detected signature for surface reflection effects.

- When analyzing experimental data, HYDROLIGHT can be run repeatedly with different water optical properties and boundary conditions, to see how particular features of the data are related to various physical processes or features in the water body, to substance concentrations, or to boundary or other external environmental effects. Such simulations can be valuable in formulating hypotheses about the causes of various features in the data.

- HYDROLIGHT can be used to simulate optical signatures for the purpose of evaluating proposed remote-sensing algorithms for their applicability to different environments or for examining the sensitivity of algorithms to simulated noise in the signature.

- HYDROLIGHT can be used to characterize the background environment in an image. When attempting to extract information about an object in the scene, all of the radiance from the natural environment may be considered noise, with the radiance from the object being the signal. The model can then be used to compute and remove the environmental contribution to the image.

- HYDROLIGHT can be run with historical (climatological) or modeled input data to provide estimates about the marine optical environment during times when remotely or in-situ sensed data are not available.

- HYDROLIGHT can form the core of implicit inverse models. Inverse models attempt to deduce the water IOP's from light field measurements. Implicit inverse models do so by generating a sequence of "forward" or "direct" solutions, in which the light field is predicted from assumed IOP's. HYDROLIGHT is a forward model and can be used to generate the sequence of forward solutions leading to the inverse solution.

The output from HYDROLIGHT can take many forms: water-leaving radiances for remote-sensing applications, irradiances for use in computing biological primary productivity or water heating rates, in-water apparent optical properties (such as K functions) for lidar bathymetry applications, or ambient light field data as may be relevant to underwater visibility or optical communications applications.

2.3 Software Requirements

The HYDROLIGHT 3.0 source code is written entirely in FORTRAN 77. Previous versions of the HYDROLIGHT code have been run on a variety of computers including CDC and Cray mainframes, Sun and Silicon Graphics workstations, and an IBM 486 PC clone with the UNIX operating system. Version 3.0 should be readily portable to any machine with a FORTRAN 77 (or later) compiler and virtual memory. The requirement of virtual memory ensures that the user does not have to worry about array sizes.

The HYDROLIGHT 3.0 source code as distributed consists of the following parts:

- A core of main programs and subroutines. Only the most sophisticated users would even contemplate tampering with the highly mathematical core routines.

- A collection of standardized subroutines. These routines provide the core program with information about the absorbing and scattering properties of the particular water body being simulated, about the sky radiance distribution, and the like. Users will often want to write their own versions of these routines in order, for example, to read in their own measured absorption and scattering profiles or to

insert their own analytical models of the inherent optical properties. Many examples of these standardized subroutines are provided with the HYDROLIGHT code.

- A collection of public domain subroutines taken from LAPACK (Linear Algebra PACKage) and BLAS (Basic Linear Algebra Subroutines). These routines are used by the core routines for mathematical tasks such as matrix inversion and eigenvector-eigenvalue analysis. These routines were obtained via *netlib* from the Oak Ridge National Laboratory; see Dongarra and Grosse (1987).

- Example files of UNIX job control language "run files," input data, and printout. These files show the mechanics of running the HYDROLIGHT code on a UNIX system and give example output, which can be duplicated by the user to verify that the code is running properly on the user's computer.

- A small collection of plotting routines written in IDL. These routines are, strictly speaking, not a part of the HYDROLIGHT code. They are included for the convenience of users who have the IDL software package. Users may wish to discard these routines and use other software for graphical analysis of the HYDROLIGHT digital output.

In addition to the distributed code, HYDROLIGHT requires a few FORTRAN subroutines from *Numerical Recipes, Second Edition*, by Press et al. (1992). These routines cannot be distributed with the HYDROLIGHT code because of copyright restrictions, but they can be purchased at a nominal cost (see the Press et al., 1992 reference). The *Numerical Recipes* routines are used for mathematical tasks such as random number generation and solution of ordinary differential equations. The locations of these routines throughout the code are clearly marked with comment cards for the convenience of users who may wish to substitute other corresponding routines.

2.4. Computational Efficiency

HYDROLIGHT 3.0 employs mathematically sophisticated invariant imbedding techniques to solve the quad- and band-averaged RTE. Details of this solution method are given in Mobley (1994). Invariant imbedding is computationally extremely fast compared

to other solution methods such as discrete ordinates and Monte Carlo simulation. Computation time is almost independent of the depth variability of the inherent optical properties (whereas a discrete ordinates model, which resolves the depth structure as N homogeneous layers, takes N times as long to run for stratified water as for homogeneous water). Computation time depends linearly on the depth to which the radiance is desired (whereas Monte Carlo computation times increase exponentially with depth). All quantities are computed with equal accuracy, and there is no statistical noise in the results. Monte Carlo models suffer from statistical noise, and quantities such as radiance contain more statistical noise than quantities such as irradiance, because the simulated photons must be partitioned into more directional bins (such as the quads) when computing radiances. The water-leaving radiance – the fundamental quantity in remote sensing studies – is very time consuming to compute with Monte Carlo simulations because so few incident photons are backscattered into upward directions.

The computational advantages of invariant imbedding for one-dimensional geometries have long been recognized in other applications of radiative transfer theory, such as atmospheric and nuclear-reactor physics. The model comparison study reported in Mobley et al. (1993) highlighted its advantages in optical oceanography. Invariant imbedding most handsomely repays the efforts required for its understanding and coding when it is applied to simulations requiring full radiance distributions, calculations of backscattered radiance (as in remote sensing studies), calculations at great optics depths, or calculations in water bodies whose IOP's vary rapidly with depth (such as is often observed with recently developed instruments capable of measuring absorption and attenuation *in situ* with vertical resolution on the order of centimeters).

Previous applications of HYDROLIGHT have been to problems such as biological productivity of the oceans, in which case it is necessary to compute the radiance distribution throughout the euphotic zone (roughly the upper five diffuse attenuation depths of the water). Such a run takes a few minutes of computer time for each wavelength, when run on a Sun SPARCstation 2. In a remote sensing study, it may be sufficient to compute the radiance only within the upper one or two diffuse attenuation depths, because most of the water-leaving radiance comes from the near-surface water. The run time is then reduced proportionately. Running the model for N wavelengths takes N times as long as for one wavelength. Experience shows that practical applications of HYDROLIGHT require from a few minutes of SPARCstation 2 time for a run at a single wavelength to a few hours of SPARCstation 2 time for a run with 5 nm bandwidths over the 350-700 nm region. Modern workstations are at least an order of magnitude faster than a SPARCstation

2. Therefore, making multiple runs for the analysis of hyperspectral remote-sensing data is well within the computational capabilities of desktop computers.

The emphasis in writing the HYDROLIGHT code was to ensure that it gives correct results. Relatively little time has been spent on investigating algorithms for speeding up the computations themselves. Most of the computer time is spent in solving a set (typically hundreds) of ordinary differential equations [Eqs. (8.74)-(8.85) of *Light and Water*]. This solution is carried out using a high-order Runge-Kutta algorithm contained in the *Numerical Recipes* routines. It is possible that the Runge-Kutta routine could be replaced with a more sophisticated algorithm, with perhaps a great improvement in the overall run speed, but time has not permitted such investigations.

2.5 Running the Code

The routines of the HYDROLIGHT 3.0 code are grouped into two parts. Part 1 of the code performs certain calculations associated with the air-water surface boundary conditions. These calculations depend only on the wind speed and on the chosen quad partitioning. In particular, note that these surface calculations are independent of the sky radiance distribution and of the water inherent optical properties. Therefore, Part 1 needs to be run only once for a given wind speed and quad partitioning. The results of Part 1 are saved for repeated use by Part 2 of the code, which performs the remainder of the computations. All of the environmental input to the HYDROLIGHT code, except for the wind speed, is read in by the routines of Part 2.

Part 2 of the code runs in either of two modes – the discretization mode or the solution mode. One consequence of the discretization of the RTE into quads of finite solid angles is that the scattering phase function must be correspondingly discretized. Recall that the phase function $\tilde{\beta}(\theta',\phi' \rightarrow \theta,\phi)$ gives the probability per unit solid angle that a photon originally traveling in direction (θ',ϕ'), if elastically scattered, will be scattered into direction (θ,ϕ). Phase functions for natural waters depend only on the angle ψ between (θ',ϕ') and (θ,ϕ); we can therefore write the phase function as $\tilde{\beta}(\theta',\phi' \rightarrow \theta,\phi) = \tilde{\beta}(\psi)$. The angle ψ is called the scattering angle. The discretized version of the phase function, $\tilde{\beta}(r,s \rightarrow u,v)$, gives the probability that photons originally heading anywhere into quad Q_{rs} will be scattered somewhere into quad Q_{uv}. The discretization mode of Part 2 computes $\tilde{\beta}(r,s \rightarrow u,v)$ for all pairs of quads, for a given phase function $\tilde{\beta}(\psi)$. For phase functions that are highly peaked in the forward direction, such as the well-known phase functions determined by Petzold, the discretization calculations can be computationally time

consuming. (The details of these discretization calculations are given in Section 8.2 of *Light and Water*.) However, these discretization calculations need be performed only once for a given phase function and quad partitioning. The discretized version of a given phase function is saved for repeated use when running Part 2 in solution mode.

After all needed phase functions have been discretized, Part 2 is run in solution mode. In solution mode, Part 2 takes input from several sources: a file of surface information (from Part 1), one or more files of discretized phase functions, one or more routines defining the IOP's of the water body, a file specifying the output desired, and perhaps files containing information about the sky radiance distribution or the bottom boundary reflectance. Using this input, Part 2 then completes the solution of the discretized RTE and creates the final output from the HYDROLIGHT model.

The normal sequence of events in a modeling study using HYDROLIGHT 3.0 is then as follows:

(1) Select a quad partitioning and then run Part 1 once for each wind speed of interest.

(2) Run Part 2 in discretization mode once for each phase function of interest.

(3) Define a unique simulation by selecting the IOP's of the water body, the sky radiance, the bottom depth and type, and the wind speed. Attach the appropriate files from steps (1) and (2) and run Part 2 in solution mode.

(4) Analyze the output from step (3).

(5) Return to step (3) and repeat the solution process using, perhaps, a new wind speed, a different sun location, different IOP's, or a different bottom condition.

It is often possible to cycle through steps (3) and (4) in a nearly automated fashion in order to carry out a detailed study of how the marine light field depends on some quantity of interest, such as the sun's location or the IOP's of the water body. Note that it is necessary to return to steps (1) or (2) only if a brand new quad partitioning, wind speed, or phase function is required in the study.

Section 3 of this report gives the details of how to run Part 1 of HYDROLIGHT 3.0. Section 4 describes how to run Part 2 of the model.

2.6 History

Development of the HYDROLIGHT quad-averaging algorithms and associated computer code began in 1979 when the author began working with the late Rudolph Preisendorfer. Work continued sporadically as time permitted, and version 1.0 of the code was released in 1988. The original code, called the Natural Hydrosol Model (something of a misnomer), is documented in detail in Preisendorfer and Mobley (1985), Mobley (1988), and Mobley and Preisendorfer (1988). That code was rewritten several times as it was ported from one computer to another, and as computer features such as virtual memory became available. The most recent (undocumented) version 2.1 was created in 1992 when the code was ported from a CDC mainframe to a UNIX workstation. These previous versions of the code all were based on the monochromatic RTE; thus the solutions could not simulate the effects of inelastic scatter, nor did the model include internal source terms (such as for bioluminescing organisms).

Because of the potential applications of the model to hyperspectral remote sensing of coastal waters, in 1994 the Office of Naval Research funded a major rewrite of the code; the result is the present version 3.0, renamed HYDROLIGHT. The main features of version 3.0 are the inclusion of inelastic scattering effects and internal sources, rewriting some of the ancient FORTRAN left over from the early 1980's, renaming the variables in the code to correspond to the modern notation seen in Mobley (1994), rewriting the comments in the code to refer to Mobley (1994), and writing a new users' guide.

3.0 RUNNING PART 1

This section of the users' guide describes in detail how to run Part 1 of HYDROLIGHT. The task of Part 1 is to compute four bi-directional radiance reflectance and transmittance functions that specify the optical properties of the air-water surface *itself*. These functions describe how quad-averaged radiance traveling in any direction is reflected and transmitted by the wind-blown water surface. These are the functions described in Section 4.7 of *Light and Water* and seen in their final form in Eq. (4.74) therein.

Although HYDROLIGHT uses invariant imbedding techniques to solve the RTE in Part 2, Part 1 uses Monte Carlo ray tracing to estimate the four surface reflectance and transmittance functions. This brute-force numerical method is employed simply because it is the only mathematically tractable way to simulate the radiative properties of random sea surfaces.

3.1 Decisions To Be Made

The only environmental variable that must be specified in Part 1 is the wind speed U, which is in meters per second at an anemometer height of 12.5 m. U is used to define the statistical properties of the random capillary wave surface, as seen in Eq. (4.32) of *Light and Water*. The remaining input to Part 1 defines the partitioning of (θ,ϕ) into quads and specifies how many rays are to be traced in the Monte Carlo computation of the radiance transfer functions.

As mentioned in Section 2.1, the quad partitioning divides the infinite number of directions (θ,ϕ), $0 \le \theta \le \pi$, $0 \le \phi < 2\pi$, into a finite number of regions bounded by lines of constant θ and constant ϕ, plus two polar caps. There are M "bands" of quads in the θ direction, counting the two polar caps, and N bands in the ϕ direction. The lines of constant θ can be arbitrarily spaced, but the lines of constant ϕ are equally spaced.

Figure 1 shows four examples of quad partitioning. Figure 1(a) shows an "equal $\Delta\theta$" partitioning with $M = 20$, $N = 24$, and an equal spacing $\Delta\theta$ of the lines of constant θ, except for the polar caps, which have a half angle of $\Delta\theta/2$. The spacing of the ϕ lines is $\Delta\phi = 360°/24 = 15°$. Figure 1(b) shows an "equal solid angle" partitioning with the same number of quads, but now each quad has the same solid angle $\Delta\Omega = \Delta\mu\Delta\phi$, where $\mu = \cos\theta$. Figure 1(c) shows an "ad hoc" partitioning in which the lines of constant θ are closely spaced near the "equator." This partitioning could be used to get increased resolution in θ for solar angles very near the horizon. Other ad hoc partitionings could be used, for example, to force the $\Delta\theta$ values to be the same as the field of view of a particular sensor. Figure 1(d) shows an equal $\Delta\theta$ partitioning as in Fig. 1(a), but with $M = 30$ and $N = 36$.

The quad partitioning determines the directional resolution of computed radiances. Recall that the quantity computed by HYDROLIGHT is the actual radiance averaged over each quad. Thus the quads act as "frosted glass windows" that homogenize the radiance within each quad. HYDROLIGHT therefore cannot distinguish, for example, between two solar positions that are in the same quad; it can distinguish only between solar positions in different quads. For most oceanographic applications, changing the sun's zenith angle from, say, 20° to 22° to 24° will have a negligible effect on the marine light field. Such small changes may lie within one quad. However, changing the solar zenith angle from 20° to 30° to 40° will have a measurable effect on the light field; such large changes in the solar θ value also will be resolvable by a quad resolution like that of Fig. 1(a). An equal $\Delta\theta$ partitioning like that of Fig. 1(a) is therefore recommended for routine calculations.

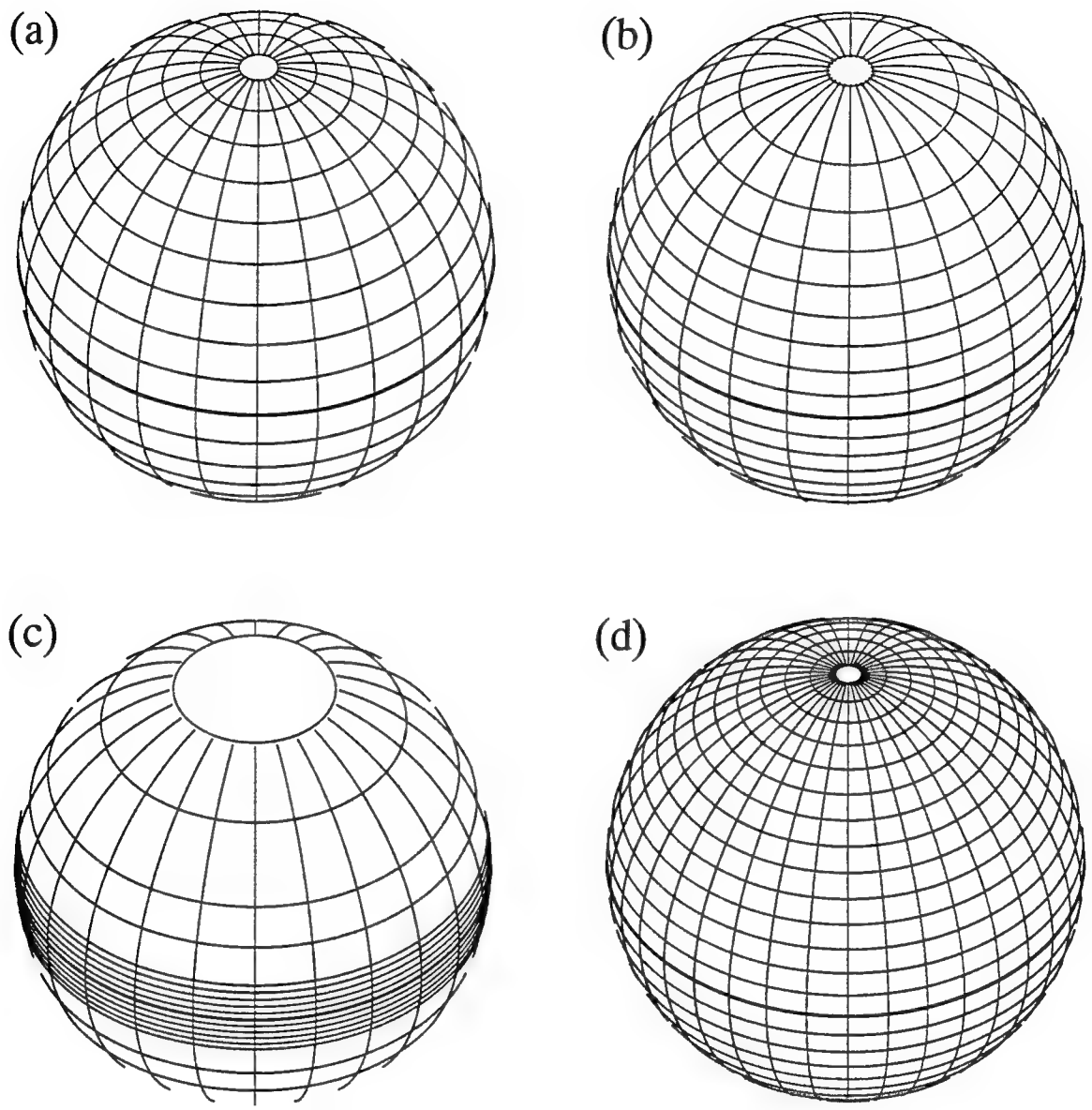

Figure 1. Four examples of quad partitioning. Panel (a) shows an equal $\Delta\theta$ partitioning with $M = 20$, $N = 24$; panel (b) shows an equal solid angle partitioning with $M = 20$, $N = 24$; panel (c) shows an ad hoc partitioning $M = 20$, $N = 24$; panel (d) shows an equal $\Delta\theta$ partitioning with $M = 30$ and $N = 36$.

Once the wind speed and quad partitioning have been chosen, the user must specify how many "rays" (simulated photon packets) should be traced for each quad. We can visualize the process as sending rays from a particular quad toward the sea surface, which is regarded as lying at the center of a sphere of quads seen in Fig. 1. When a "parent" (original) ray reaches the sea surface, it reflects and refracts off of the wavy, wind-blown sea surface. The reflected and refracted "daughter" rays can undergo further interactions with the sea surface (multiple scattering), but they eventually travel away from the surface in random directions and pass through various quads, where their energy is tallied. This tally of ray energies for all pairs of quads is the basis for computing the air-water-surface radiance transfer functions. Section 4.8 of *Light and Water* gives specific examples of this ray-tracing process.

The parent rays are started from randomly chosen directions within a particular "input" quad, and the daughter rays can end up traveling into any "output" quad. It is necessary to trace enough parent rays from each input quad to get a statistically stable distribution of rays connecting the input and output quads. The more rays connecting a pair of quads, the smaller is the statistical noise in the estimate of the radiance transfer functions for that pair of quads. Output quads near to the specular reflection and transmission directions of the input quad accumulate rays quickly (the specular directions are the directions of the reflected and refracted rays if the sea surface is perfectly level, which corresponds to a wind speed of zero). Output quads at large angles away from the specular directions accumulate rays slowly, because scattering is much more likely to occur in near-specular directions, especially at low wind speeds.

It is difficult to estimate *a priori* how many rays should be traced, because the answer depends on the quad solid angles and wind speed, and on the acceptable level of Monte Carlo noise in the computed transfer functions. The user can check the acceptability (for his or her accuracy requirements) of a given number of rays as follows. Make multiple runs of Part 1 with different seeds for the random number generator (as described in Section 3.2, below), but with all other variables being held constant. Request printout of the various transfer functions (via setting *idbug* = 1 or 2, as described below). Comparison of the corresponding array elements for different seeds then gives an idea of the variability in the array elements owing to Monte Carlo noise. The array elements of largest magnitude will display very little variability, because they result from many rays being tallied. Accurate computations of the radiance distribution will be obtained in Part 2 if the large- and intermediate-sized array elements change by a few percent or less. The smallest array elements (of order 10^{-6} or less, relative to the largest elements) will always

display the greatest variability, because they result from only a few tallied rays. It is acceptable to have high variability in the smallest array elements, because they represent a negligible part of the total energy being reflected and refracted by the sea surface. Experience based on such simulations shows that for typical quad partitionings like those of Figs. 1(a), 1(b), or 1(d), 500 rays per input quad is the minimum acceptable number for low wind speeds, $U \leq 2$ m s^{-1}; 2000 rays is the minimum for moderate wind speeds, $U \sim 5$ m s^{-1}; and 5000 rays is the minimum for high wind speeds, $U \geq 10$ m s^{-1}. Tracing even more rays is computationally reasonable, since computer time is not critical for these one-time calculations.

3.2 Detailed Description of Run-Time Input

All input to Part 1 is read from FORTRAN logical unit 5 (UNIX "standard input") by a subroutine named inishl1, which is found on file inishl1.f in the HYDROLIGHT source code. It is usually convenient to place the input records in an ASCII file, and then link that file to the run as standard input at run time. There are three input records, defined as follows:

RECORD 1.
This record gives a descriptive title to identify the printout. The title contains a maximum of 120 alphanumeric characters.

RECORD 2.
This record defines the wind speed and quad partitioning. The record is free-format and has the form

U, M, N, iqpart

where

U = the wind speed in m s^{-1} at 12.5 m elevation
M = the number of θ (μ) bands in 0 to π (180°); M MUST be even
N = the number of ϕ bands in 0 to 2π (360°); N MUST be a multiple of 4
iqpart defines the kind of quad partitioning, as follows:
if *iqpart* = 1, all quads have equal $\Delta\theta$ values
if *iqpart* = 2, all quads have equal solid angles
if *iqpart* = 3, the quads have an ad hoc θ partition, defined by
user-written subroutine adhoc, which is found on file adhoc.f.

RECORD 3.

This record defines the amount of printout and the ray-tracing parameters. It is free-format and has the form

idbug, nray, iseed

where

idbug = 0, 1, or 2 for increasing printout amounts; use 0 for minimum output

nray = the number of initial rays to be traced for EACH quad

iseed = the seed for random number generation; $1 \le iseed \le 2^{31} - 1 = 2147483647$

Several other variables are required for running Part 1; these variables are set to default values in subroutine inishl1. For example, the index of refraction of the water is given a default value of 1.34, and the parameters used in the Cox-Munk wave-slope wind-speed law for capillary waves are set to their empirical values. Such parameters can be changed by the user if necessary for special applications of HYDROLIGHT.

Array dimensioning in the distributed code supports values of $M \le 30$ and $N \le 36$, the values used in generating Fig. 1(d). Such a number of quads is more than adequate for most oceanographic applications of HYDROLIGHT. Users who need more quads must increase the dimension parameters *mxmu* (the maximum value of $M/2$) and *mxphi* (the maximum value of N) throughout both Parts 1 and 2 of the code. See the comments in inishl1.

3.3 Required Routines

In addition to the core code, Part 1 requires three routines from *Numerical Recipes, Second Edition*:

gasdev — generates a sequence of normally distributed pseudo-random numbers with zero mean and unit variance.

indexx — generates an index array *indx*(1:*n*) for a given array *arr*(1:*n*) such that *arr*(*indx*(*j*)) is in ascending order for $j = 1, 2, ..., n$.

ran1 — generates a sequence of uniformly distributed pseudo-random numbers on the interval 0 to 1.

See *Numerical Recipes, Second Edition* for detailed descriptions of these routines. The user can replace these routines with equivalent ones.

Users wishing to define ad hoc quad partitionings, as illustrated in Fig. 1(c), can do so by modifying subroutine adhoc, which is supplied with the core code. The example adhoc was used in generating Fig. 1(c).

3.4 Example Output

The distributed source code also contains ASCII files named Make1, Run1, Input1, and Output1. These files were used to generate an example set of output from Part 1, using the author's Sun SPARCstation 2 workstation. The user should be able to duplicate this output after installation of the HYDROLIGHT code on his or her computer system. The files Make1 and Run1 are specific to the UNIX operating system. Users with other operating systems will have to replace these files with corresponding files of job control language for their computer. For the most part, the following discussion of the example run presumes that the reader has obtained the source code and can refer to the various ASCII files as needed in order to follow along with the discussion.

Make1 is a UNIX "make file" whose purpose is to compile all of the FORTRAN source code for Part 1 and link the compiled routines into an "executable" file. Note that the user must supply the three routines from *Numerical Recipes*. On a UNIX system, Make1 is invoked by the command <make -f Make1>. (Here, the angle brackets <...> delimit the actual command.)

Input1 is the file containing the three input records described in Section 3.2. The entire contents of this example input file are as follows:

```
Example run of Part 1, for the Users' Guide
5.0 20 24 1
1 2000 4992275
```

Run1 is a file of UNIX job control language that links Input1 as standard input, and sends the "printout" (FORTRAN unit 6, or UNIX "standard output") to a file named Output1. Part 1 always writes its binary output to a file named rtspec. After execution, file rtspec is saved with a descriptive name. In the present case it is called rtspec.u5.20x24, which reminds us that it is file rtspec for a 5 m s^{-1} wind speed (U = 5) and a quad partitioning with 20 and 24 quads in the θ and ϕ directions, respectively. This binary file will be read as input by Part 2 of the HYDROLIGHT model. Binary files such

as rtspec are not distributed with the HYDROLIGHT code because they are machine dependent.

The printout on distributed file Output1 contains output as generated with *idbug* = 1, plus several annotations to help the reader in understanding the output. As mentioned above, setting *idbug* = 1 generates extra output that can be used for estimating Monte Carlo noise. The extra printout also contains various numerical checks on the computed quantities. These checks are intended primarily for debugging if modifications are made to the code; they are not made if *idbug* = 0. The run times shown in the printout are "wall clock" times, not true execution times. These times can vary for identical runs, depending on the computation load on the computer while the HYDROLIGHT code is running. Nevertheless, these times are useful for estimating the consequences, say, of greatly increasing the number of rays traced.

4. RUNNING PART 2

Part 2 performs most of the work of the HYDROLIGHT model. Part 2 takes the information about the sea-surface, saved from a previous run of Part 1, and adds information on the IOP's of the water body, on the incident sky radiance distribution, and on the bottom boundary condition. This information together completely defines a physical situation for which the quad-averaged radiative transfer equation has a unique solution. Part 2 then solves the quad-averaged RTE and generates the final output of the HYDROLIGHT model.

The computations in Part 2 are entirely analytical; no Monte Carlo simulations are used in solving the RTE itself.

4.1 Decisions To Be Made

With the exception of the wind speed, all of the environmental variables are specified in Part 2. Other information, such as the wavelength bands to be used and the depths at which output should be saved for later analysis, also must be provided to Part 2. Consequently, the input to Part 2 is considerably more complicated than was the input to Part 1. Some of the needed information is easily specified in an input file of a few records, whereas other information may require the user to write FORTRAN subroutines to read and process user-supplied data before passing the needed information to

HYDROLIGHT on a specified format. The distributed HYDROLIGHT code contains many examples of how to supply this information to the core code.

Note that the HYDROLIGHT model *per se* is a radiative transfer model, not a model of oceanic optical properties. The user must supply the IOP's to the HYDROLIGHT core code. (Indeed, the HYDROLIGHT model is not even restricted to the oceanic setting, although that is our interest here. If you supply HYDROLIGHT with the optical properties of orange paint, for example, then HYDROLIGHT will happily solve for the radiance distribution within and leaving the paint.) Therefore, prior to running Part 2, the user must decide what IOP's to use, what sky radiance to use, what bottom boundary condition to use, and so on. The next pages discuss these decisions in detail.

4.1.1 Defining the Inherent Optical Properties

The IOP's are provided to the HYDROLIGHT core code by two user-written routines, one for the phase function $\tilde{\beta}$ and one for the absorption and scattering coefficients a and b. Several examples of such routines are provided with the distributed HYDROLIGHT code. Users can use these examples as provided or regard them as templates for writing their own versions of the needed routines.

Routine phasef

This FORTRAN function subprogram returns the value of the *normalized* phase function $\tilde{\beta}$ for any given value of the *cosine of the scattering angle* ψ. The units of $\tilde{\beta}$ are sr^{-1}. The HYDROLIGHT core code (in particular, routine qaphas) repeatedly invokes this routine via statements of the form

x = phasef(*cospsi*)

where *cospsi* is a value of $\cos(\psi)$. The distributed HYDROLIGHT code contains four examples of phasef on files named as follows:

pfconst.f — returns the value $\tilde{\beta} = 1/4\pi$ sr^{-1}, the value of $\tilde{\beta}$ for isotropic scattering, regardless of the value of *cospsi*.

pfothg.f — returns $\tilde{\beta}$ as defined by a one-term Henyey-Greenstein phase function, Eq. (3.34) of *Light and Water*. The g parameter is set in the routine.

pfpart.f returns an average "particle" phase function. This phase function is defined from Petzold's data as described in *Light and Water*, pages 109-112.

pfpure.f returns $\tilde{\beta}$ for pure sea water, Eq. (3.30) of *Light and Water*.

The user can take these phase functions in their existing forms or can modify them in an obvious fashion to define other phase functions. The only constraints on modifying the examples are that the user's phase function must satisfy the normalization condition

$$2\pi \int_0^{\pi} \tilde{\beta}(\psi) \sin\psi \, d\psi = 1 ,$$

and that the routine phasef return a value for any allowed value of *cospsi*, that is, for $-1 \le \cos\psi \le 1$, which corresponds to $0 \le \psi \le \pi$.

Routine abscat

This subroutine defines the absorption and scattering coefficients of the water body, $a(z,\lambda)$ and $b(z,\lambda)$, respectively. The units of a and b are m^{-1}, z is usually in meters, and λ is in nanometers. Numerous routines in the HYDROLIGHT core code invoke routine abscat via calls of the form

call abscat(*z,acomp,bcomp,atotal,btotal*).

Here z is the current value of the depth. The current value of the wavelength λ is passed to abscat via common block cmisc. Variables *acomp* and *bcomp* are one-dimensional arrays that return the absorption and scattering coefficients for each of the components of the water. For example, suppose (as we shall in Example 4, below) that the routine models the water body as having *ncomp* = 3 components: pure water for component 1, phytoplankton for component 2, and dissolved substances for component 3. Then *acomp*(1) is the absorption by pure water, *acomp*(2) is the absorption by phytoplankton, and *acomp*(3) is the absorption by dissolved substances; *bcomp*(i) is defined correspondingly for scattering by the i^{th} component. Variables *atotal* and *btotal* are the total absorption and scattering at the current z and λ, that is, the sums of *acomp*(i) and *bcomp*(i) for i = 1, ..., *ncomp*. Dissolved substances usually are assumed to be nonscattering, thus *bcomp*(3) = 0

in the above example. If abscat is reading measured values of total a and total b, then *ncomp* = 1 and *acomp*(1) = *atotal*, and so on.

Array dimensioning in the distributed HYDROLIGHT code supports up to *ncomp* = 10 components in a model of the water IOP's. Users needing more that 10 components must increase parameter *mxcomp* throughout the routines of Part 2.

The partitioning of *btotal* into component contributions is used to weight the corresponding component phase functions as seen in Eq. (3.13) of *Light and Water*. Thus if the user decides to call pure water component 1 and phytoplankton component 2, then the phase function 1 *must* be for pure water (the $\tilde{\beta}$ from file pfpure.f) and phase function 2 *must* be for particles (e.g., the $\tilde{\beta}$ defined on file pfpart.f). We shall see below how this information is given to the HYDROLIGHT code at run time.

The distributed HYDROLIGHT code contains five examples of abscat on files named as follows:

abcase1.f — computes a and b from a two-component (pure water plus chlorophyll-bearing particles) bio-optical model of Case 1 water. This version of abscat also calls a function subroutine chlz(z) (on file chlz.f) that returns the chlorophyll concentration at the current depth z.

abcase2.f — computes a and b from a three-component (pure water plus chlorophyll-bearing particles plus dissolved substances) model of Case 2 water. This version of abscat calls chlz(z) and a function subroutine acdom(z,λ), which returns the absorption at z and λ due to CDOM. It is straightforward to add other components such as detritus or mineral particles to this routine.

abconst.f — returns values of a and b that are independent of depth. This routine is useful for single-wavelength runs in which the a and b values are read at run time from file Input2, as described in Section 4.2.

abmeas.f — computes a and b from a particular set of actual measured values of total a and beam attenuation c (which is found on file Data.ex3). The scattering coefficient is obtained from $b = c - a$. Furthermore, the measurements were made only at discrete depths, so the

measured data are first fit with a cubic spline, which is then used to compute a and b at any depth z. This routine is an example of how abscat can read and process user-supplied data in order to obtain the a and b values needed by HYDROLIGHT.

abpure.f returns the a and b values of pure sea water (the well-known Smith and Baker data).

The two routines phasef and abscat together define the IOP's of the water column, although those routines may call others routines such as chlz and acdom.

4.1.2 Defining the Sky Radiance Distribution

The next decision is to define the sky radiance distribution.

Routine qasky

This subroutine first defines the directionally continuous sky radiance distribution and then computes the quad-averaged sky radiance, which is the input actually needed by HYDROLIGHT. The term "sky radiance" means the radiance incident onto the water surface from the sun, moon, and clouds, as well as from the background sky itself. The distributed code includes two versions of qasky, which are likely to be adequate for most applications of the HYDROLIGHT model. However, users can modify the supplied versions of qasky, for example, in order to read in actual measured sky radiances. The two versions of qasky are on the following files:

qacardsky.f defines the sky radiance distribution using the cardioidal radiance distribution of *Light and Water*, Eq. (4.50). This simple analytic radiance distribution is useful for single-wavelength studies of idealized situations, such as a uniform sky or a point sun in a black sky.

qarealsky.f defines the sky radiance distribution by piecing together the semi-empirical models of Gregg and Carder (1990), Harrison and Coombes (1988), and Kasten and Czeplak (1980). These models

give both reasonably accurate angular distributions of the sky radiance and accurate spectral irradiance values for a variety of atmospheric conditions. This version of qasky should be used for routine calculations.

Both versions of qasky require various parameters as input. For example, the cardioidal-sky version requires values for the total sky irradiance, the cardioidal parameter, and the solar zenith angle. The real-sky version requires, among other things, either the solar zenith angle or the Julian date, time, latitude, and longitude, from which the solar zenith angle is computed within the subroutine. These basic parameters are read at run time from file Input2, as described in Section 4.2. Other parameters such as aerosol type, relative humidity, visibility, and ozone concentration are set to default values within the routines found in file qarealsky.f. These defaults can be changed by the user to get great flexibility and realism in the modeling of sky radiance distributions.

4.1.3 Defining the Bottom Boundary

The HYDROLIGHT model can simulate both infinitely deep and finite-depth water bodies. Let m be the "maximum depth of interest" in a run, that is, the maximum depth at which we wish to obtain output from the model. For a study in the open ocean, we might take m = 100 m, even though the water is optically infinitely deep. In this case, the model assumes that the water is homogeneous below depth m and has the same IOP's as are given by abscat at $z = m$. (Note that the water column between depth 0, the mean sea surface, and depth m generally has depth-dependent IOP's.) The model automatically computes the bi-directional radiance reflectance of the infinitely deep, homogeneous layer of water below depth m and applies this reflectance as the bottom boundary condition *at* depth m. Section 9.5 of *Light and Water* describes these calculations.

Routine rbottom

In the finite-depth case, a physical bottom is placed at depth m. The physical bottom is taken to be an opaque, Lambertian reflecting surface with a given *irradiance* reflectance. Equation (4.81) of *Light and Water* then gives the needed bi-directional *radiance* reflectance of the physical bottom. The function subroutine rbottom found on file botmbc.f contains spectral irradiance reflectance data for several bottom types such as

clean coral sand and green, brown, and red algaes. These reflectances are derived from data in Maritorena, Morel, and Gentili (1994). Users wishing to use their own bottom reflectance data can add the data to routine rbottom in an obvious fashion.

4.1.4 Defining the Output Depths

A defining characteristic of the HYDROLIGHT model is that it solves the RTE for *any* depth dependence of the IOP's. This solution process is automatic as the core code integrates the Riccati equations described in *Light and Water*, Section 8.7. The core integration routines repeatedly call abscat to obtain the absorption and scattering coefficients as functions of depth. The self-monitoring integration algorithms take sufficiently small steps in the integration process to maintain high accuracy in the final solution radiances. There is no restriction on how complicated the depth dependence of a and b can be. HYDROLIGHT is emphatically *not* a "layered" model (as is discrete ordinates), which approximates the depth dependence as a stack of homogeneous layers.

Although HYDROLIGHT automatically solves the RTE with arbitrarily fine depth resolution in the IOP's, it *saves* the computed radiances only at a *preselected* set of depths z_k, $k = 1, 2, ..., K$. These *output depths* must be chosen along with the other input to Part 2. The output depths can be arbitrarily spaced in order to get detailed output in the regions of greatest interest, such as near the sea surface or near strong gradients in the IOP's, and less output in regions where there is little "fine structure." *It is very important to note that the solution of the RTE is entirely independent of where output is to be saved for later analysis.* For example, we might request output at one-meter depth intervals, that is, $z_1 = 0$, $z_2 = 1.0$ m, $z_3 = 2.0$ m, ..., $z_6 = 5.0$ m, ..., and so on. On the other hand, we might request output at $z_1 = 0$, $z_2 = 5.0$ m, ..., and so on. The output at 5.0 will be exactly the same in both cases; the only difference is that we will have additional output between 0 and 5.0 m in the first case. In particular, HYDROLIGHT does *not* solve the RTE with "one-meter depth resolution" in the first case and with "five-meter resolution" in the second case.

The accurate computation of K-functions imposes an additional constraint on the selection of output depths. Consider, for example, K_d, the diffuse attenuation coefficient for downwelling plane irradiance E_d. The radiance is saved at each requested output depth z_k, and from this the irradiance E_d is computed at each z_k. The value of K_d is then estimated from

$$K_d \doteq -\frac{1}{E_d}\frac{dE_d}{dz} \approx -\frac{1}{0.5\,[E_d(z_k) + E_d(z_{k+1})]}\,\frac{E_d(z_{k+1}) - E_d(z_k)]}{z_{k+1} - z_k}. \qquad (2)$$

The value of K_d so computed is the *average* value of K_d over the depth range z_k to z_{k+1}. Many oceanographers need depth profiles of K_d with as much detail as possible in the depth variability of K_d. However, the finite-difference approximation is an accurate estimate of K_d *at the midpoint* $\frac{1}{2}(z_k + z_{k+1})$ only if the two depths z_k and z_{k+1} are "close together," even if the irradiances are computed with perfect accuracy. *The default printout from the model therefore assumes that the output depths have been requested at closely spaced pairs of depths.*

The requested output depths therefore could be $z_1 = 0.00$, $z_2 = 0.01$, $z_3 = 1.00$, $z_4 = 1.01$, ..., $z_k = 10.00$, $z_{k+1} = 10.01$, ... The model then gives printout of irradiances and their K-functions only at the odd numbered depths, z_1, z_3, ..., z_k, ..., z_{K-1}. [Note that $K_d(z_k) \approx K_d[\frac{1}{2}(z_k + z_{k+1})]$ to very good accuracy when z_k and z_{k+1} are very close together.] The even numbered depths z_2, z_4, ..., z_{k+1}, ..., z_K are used only for computing K functions and are not shown in the default printout. Picking depths only 0.01 m apart exceeds the ability of oceanographic instruments to measure the corresponding changes in the light field, but such closely spaced values gives excellent depth resolution of K profiles in the numerical model.

The printout scheme just described is chosen as the default because of its convenience for most oceanographers. Also by default, the radiance distribution is not printed out because of the huge amount of data involved. However, other printout schemes are available, for example, printout at all depths or printout only at user-selected depths, and printout of all or selected parts of the radiance distribution. The user can easily select these other printout schemes by changing the values of *iprad, ipirad,* or *ipkfcn* in routine readin2, as described there.

The dimensioning in the distributed code supports output at $K = 100$ depths, for example, at 50 pairs of closely spaced depths. The parameter *mxz* must be increased throughout the code if output at more depths is needed.

Most oceanographers will wish to request output at *geometric* depths z_k, measured in meters, for ease of comparison with observational data. However, the output depths also can be specified as dimensionless *optical* depths ζ_k. This option is available only if the model is being run at just one wavelength, because the wavelength dependence of the IOP's generally makes a given optical depth correspond to different geometric depths (different physical locations in the water column) at different wavelengths. Computations in terms of optical depths are convenient for general monochromatic radiative transfer

studies, as opposed to specific oceanographic studies. Note that if the model is being run with optical depths, then routines such as abscat must be written to accept optical depth as input.

4.1.5 Defining the Wavelength Bands

The various data sets built in to HYDROLIGHT give it the ability to run anywhere in the wavelength domain from 350 to 700 nm, which is of interest in optical oceanography.

There are two options for specifying the needed wavelength information. The first is to run HYDROLIGHT at a single wavelength. HYDROLIGHT then solves the *monochromatic* RTE with all IOP's, sky radiances, and the like taken equal to their values at the specified wavelength. Note that some routines, for example the version of qasky found on file qarealsky.f, have built-in data that are accurate to 1 nm resolution. Other routines, for example the absorption coefficients for pure water and chlorophyll found on file abcase1.f, may have data built in at 5 nm or 10 nm resolution. Those routines linearly interpolate if necessary to obtain IOP's at the requested wavelength. A run at a single wavelength can include an internal source term at that wavelength. Such a source term can represent a bioluminescing layer, for example. The output depths can be either geometric or optical depths. Monochromatic runs are most useful for general radiative transfer studies.

The second option is to run HYDROLIGHT over a set of contiguous bands of bandwidths $\Delta\lambda_j$, $j = 1, 2, ..., J$. When this option is chosen, the model automatically averages the input sky radiance over each wavelength band. This averaging smooths out the large nanometer-to-nanometer fluctuations in the sky radiance magnitude owing to Fraunhofer lines in the solar spectrum. However, *the absorption and scattering coefficients are taken to be the values at the band centers*. In principle, $a(z,\lambda)$ and $b(z,\lambda)$ also should be averaged over each band. However, this averaging would have to be performed every time routine abscat is called with a new depth during the solution of the RTE; such averaging would be an enormous computational expense. If the band widths are of size $\Delta\lambda \le 20$ nm, then the replacement of band-averaged IOP values by band-center values will be acceptably accurate for most purposes, since IOP's do not fluctuate wildly on a nanometer scale.

Most present-day oceanographic and remote-sensing sensors have bandwidths between 5 and 20 nm. The bandwidths of a HYDROLIGHT run can be matched to these

sensor bandwidths as desired. There is no requirement that the bandwidths $\Delta\lambda_j$ be equal for different j values. If the user wishes to simulate a wider-bandwidth instrument, then HYDROLIGHT should be run with a number of smaller bandwidths. The final output for the smaller bandwidths then can be averaged to get band-averaged output for the wide band. This approach properly accounts for the wavelength variation of the IOP's within the large band.

A final consideration in the choice of wavelength bands arises in connection with inelastic scattering effects. Suppose, for example, that we wish to simulate the light field in the 450-500 nm region. If we are uninterested in inelastic-scattering effects from shorter wavelengths, then we could run HYDROLIGHT with five bands chosen as 450-460 nm, 460-470 nm, ..., 490-500 nm, for example. However, if we wish to include the contributions of fluorescence or Raman scattering to the light field in the 450-500 nm region, then HYDROLIGHT must be run for all wavelengths less than 450 nm for which there might be an inelastic-scattering contribution to the region of interest. Thus to include Raman scattering, the model should be run starting with a band from 390 to 400 nm, since wavelengths near 400 nm will Raman scatter into wavelengths near 450 nm. If CDOM fluorescence is to be included, then HYDROLIGHT should be run starting at 350 nm, because CDOM fluorescence can be excited by ultraviolet wavelengths and because CDOM fluoresces throughout the visible.

4.1.6 Inelastic Scattering and Bioluminescence

HYDROLIGHT 3.0 has the option of running with or without inelastic scattering and internal sources being included in the RTE. The inelastic scattering processes included in the model are chlorophyll fluorescence, CDOM fluorescence, and Raman scattering. The internal source usually is tailored to represent bioluminescence. If these effects are all omitted from the run, then HYDROLIGHT carries out a sequence of independent solutions of the monochromatic, source-free RTE. The solutions for different wavelength bands are then completely independent. However, if one or more of these effects are included, then the appropriate source terms are automatically added to the RTE, as described in *Light and Water*, Sections 5.14-5.16 and 8.7. In the case of inelastic scattering, the solutions in different wavelength bands are coupled by the inelastic scattering from shorter to longer wavelengths.

The inelastic-scattering and internal-source computations in some cases call upon user supplied routines, which are now described.

Routine chlz

Function subroutine chlz(z) returns the chlorophyll concentration in mg m^{-3} at any depth z. The core routines that compute chlorophyll fluorescence require this information. The same routine can be called by abscat if the chlorophyll concentration is required in a bio-optical model of a and b; recall the examples of abscat on files abcase1.f and abcase2.f. Further information needed in the chlorophyll-fluorescence calculations, such as the chlorophyll specific-absorption spectrum and the quantum efficiency, is set to default values in the routines found on files shatchl.f and wrfdisc.f. These default values can be changed if desired. File chlz.f contains an example of routine chlz.

Routine acdom

Function subroutine acdom(z,λ) returns the absorption by CDOM (in units of m^{-1}) at any depth and wavelength. The computation of CDOM fluorescence requires this information. As with chlz, routine acdom also can be used for other purposes such as the computation of total absorption; recall the version of abscat given on file abcase2.f. File acdom.f gives an example of acdom. HYDROLIGHT models CDOM fluorescence using the spectral fluorescence quantum efficiency function of *Light and Water* Eq. (5.101), as shown in *Light and Water* Fig. 5.11. This particular function is built in to routine wrfcdom on file wrfdisc.f. The user can replace this default function with another, if desired.

Because Raman scattering depends only on the water itself, it is always the same and no user-supplied information is required. Various parameter values, such as the Raman cross section, are set to default values in routines shatram on shatram.f and wrframen on file wrfdisc.f. Those values can be changed if desired.

Routine s0biolum

If bioluminescence is included as an internal source, then function subroutine s0biolum(z,λ) is called to get the spectral source strength (spectral radiant power) $S_o(z,\lambda)$ at any depth and wavelength. $S_o(z,\lambda)$ has units of W m^{-3} nm^{-1}, as seen in *Light and Water*, Eq. (5.107). An example of s0biolum is given on file s0biolum.f; this example routine can be rewritten as desired.

Some additional computational expense results from the addition of the source terms seen in Eqs. (8.74)-(8.85) of *Light and Water*. However, the main increase in computation time when inelastic scattering is included arises from the need to run the model over wavelengths shorter than the wavelengths of interest, as was illustrated in the previous subsection. In the example discussed there, where the interest was only on 450 to 500 nm, including CDOM fluorescence effects on the 450-500 nm band requires about three times the computational expense, because the model then must be run from 350 to 500 nm.

The HYDROLIGHT 3.0 model includes chlorophyll and CDOM fluorescence and bioluminescence exactly as formulated in Sections 5.15 and 5.16 of *Light and Water*. However, Raman scatter is included using an azimuthally averaged effective source term that is equivalent to the formulation seen in Appendix A of Mobley et al. (1993). This simplification allows the Raman effective source term to be computed from the scalar irradiance (as is the case for fluorescence), rather than from the full radiance distribution. The azimuthally averaged formalism yields the correct Raman contribution to irradiances, which are computed from the azimuthally averaged radiance. However, the Raman contribution to the radiance is correct only as an azimuthally averaged value.

4.1.7 Phase Function Discretization

Part 2 runs in either of two modes: discretization mode or solution mode. In discretization mode, the model quad averages the phase function $\tilde{\beta}$ defined by routine phasef. Section 8.2 of *Light and Water* describes these calculations in detail; the governing equation is Eq. (8.13). The file containing the desired phase function must be "loaded" in the makefile Make2, which creates the executable program file from all of the source code. Section 4.4 gives examples of how this is done. In solution mode, the model takes previously discretized phase functions as input and completes the solution of the RTE. A complete set of routines phasef, abscat, and qasky must be given in Make2, in order to avoid error messages from the makefile linking process. However, in discretization mode, only phasef is ever called; the loaded versions of abscat and qasky are irrelevant. Conversely, in solution mode, abscat and qasky are called, but the loaded version of phasef is irrelevant.

Only one phase function is quad averaged in a given run. A resulting file, named phase1, containing the quad-averaged phase function is then saved with a descriptive name for later use in the solution mode. This quad averaging depends only on the quad partitioning chosen in Part 1 and on the phase function selected for routine phasef. In

normal usage, a few one-time runs are made in discretization mode to discretize the phase functions of interest (such as those for pure water and particles). Numerous subsequent runs then can be made in solution mode for various combinations of IOP's, sky radiances, and so on. It is necessary to rerun Part 2 in discretization mode only if a new quad partitioning or a new phase function is needed.

The discretization calculations of *Light and Water* Eq. (8.13) require the partitioning of quads into "subquads." If the quads of Fig. 1 have solid angles of size $\Delta\Omega_{uv} = \Delta\mu_u\Delta\phi_v$, then the subquads have size $\delta\Omega_{uv} = (\Delta\mu_u/n_\mu)(\Delta\phi_v/n_\phi)$, where n_μ and n_ϕ are the numbers of subquads per quad in the μ (or θ) and ϕ directions, respectively. The values of n_μ and n_ϕ depend both on the size of $\Delta\Omega_{uv}$ and on the phase function. The underlying requirement for accurate discretization of $\tilde{\beta}(\psi)$ is that $\tilde{\beta}(\psi)$ does not vary greatly as the incident and scattered directions vary within any two subquads. Here ψ is the scattering angle between a subquad in quad Q_{rs} and a subquad in Q_{uv}. Experience shows that if the subquads are no more than a few angular degrees in size, then sufficient accuracy is obtained even for highly peaked phase functions, except for adjacent quads or when $Q_{rs} = Q_{uv}$, that is, when near-forward scattering occurs. In that case, the number of subquads must be increased, typically by a factor of ten. Thus for the quad partitioning of Fig. 1(a), which has $\Delta\theta \approx 9°$ and $\Delta\phi = 15°$, we can set $n_\mu = 3$ and $n_\phi = 4$ and achieve accurate values of $\tilde{\beta}(r,s\rightarrow u,v)$ except for quads involving near-forward scattering, where $\tilde{\beta}$ varies very rapidly with ψ. For adjacent quads, which give small values of ψ, the values of n_μ and n_ϕ are increased by a factor of *incfwd* = 10. For phase functions (like that of pure water) that are not highly peaked, we can set $n_\mu = 2$, $n_\phi = 2$, and *incfwd* = 1 with good results.

The discretization calculations contain an accuracy check based on the normalization condition $2\pi\int\tilde{\beta}(\psi)\sin\psi d\psi = 1$. If the computed value of the quad-averaged version of this equation differs from 1 by more than a few percent, then the discretization should be repeated with increased values of n_μ, n_ϕ, or *incfwd*. For phase functions that are highly peaked in the forward direction, the quad pair that achieves numerical accuracy most slowly is the forward-scattering quad pair, that is, the pair for which Q_{rs} is the same quad as Q_{uv}. In this case, almost all of the error in the discretization lies in the forward-scattering quad pair. This error is corrected in the last step of the discretization by use of *Light and Water* Eq. (8.14).

4.2 Detailed Description of Run-Time Input

The user-written subroutines described in the previous section provide much of the information needed to run Part 2. The remaining information is read in at run time. Subroutine readin2 reads a number of records from standard input (FORTRAN unit 5). The records usually are placed on a separate file (named, for example, Input2), which is linked to the program as standard input at run time. The following pages describe these input records in detail. Each of the records is free format.

RECORD 1.

This record gives a title for the run. The title can be up to 120 characters long.

RECORD 2.

This record specifies whether the model is being run in "discretization mode" or in "solution mode." The record reads a single number

ncomp

where

if *ncomp* = -1, then the model is being run in discretization mode. In this case another record of the form

nsubmu, nsubphi, incfwd

is read, where *nsubmu* = n_μ and *nsubphi* = n_ϕ are the numbers of "subquads" in the μ (or θ) and ϕ directions, as used in quad averaging the phase function; recall the discussion of Section 4.1.7. [These variables are the values of n_μ and n_ϕ in Eq. (8.13) of *Light and Water.*] *incfwd* is the factor multiplying the values of *nsubmu* and *nsubphi* when the discretization calculations involve near-forward scattering. For phase functions that are nearly independent of the scattering angle, such as the phase function for pure water, the values (*nsubmu, nsubphi, incfwd*) = (2, 2, 1) are recommended for quad partitions like those of Fig. 1(a). For phase functions that are highly peaked at small scattering angles, such as the particle phase function derived from Petzold's data, the values (*nsubmu, nsubphi, incfwd*) = (3, 4, 10) are recommended for quad partitions like those of Fig. 1(a). If *ncomp* = -1, no other records are read after this one: the run discretizes the phase function and stops.

if *ncomp* = 1, 2, ..., then the program is running in solution mode. *ncomp* number of previously discretized phase functions will be read from files nuphas1, nuphas2, and so on.

Records 3 through 7 are read only when the model is running in solution mode, that is, only if *ncomp* ≥ 1.

RECORDS 3 and 4.

Record 3 specifies whether or not internal sources and various kinds of inelastic scattering are to be included in the run. Record 4, whose form depends on Record 3, defines the wavelengths to be used in the run. Record 3 has the form

nwave, ibiolum, ichlfl, icdomfl, iraman

where

nwave is the number of wavelength BANDS at which the model is being run.

If *nwave* = 0, the run is to be made at an EXACT WAVELENGTH (i.e., at zero bands). In this case, Record 4 has the form

wavel, areset, breset

where *wavel* is the exact wavelength in nm, and *areset* and *breset* are values of *a* and *b* that can be used to override the values given by the version of abscat found on abconst.f. For use of the *a* and *b* values as given by abscat, read in negative values for *areset* and *breset*. The sky spectral radiance at the exact wavelength *wavel* will be used (with 1 nm resolution).

If *nwave* ≥ 1, the run is to be made with one or more finite wavelength bands. In this case, Record 4 has the form

waveb(1), *waveb*(2), ..., *waveb*(*nwave*+1)

where the values of *waveb*(*j*) give the *nwave*+1 WAVELENGTH BAND BOUNDARIES (in nm) for which the model is to be run. *a* and *b* values as returned by abscat at the band centers will be used. The band-averaged sky radiance will be used.

ibiolum is a flag for the inclusion/omission of bioluminescence:

If *ibiolum* = 0, there is no bioluminescence present.

If *ibiolum* = 1, the run includes bioluminescence; routine s0biolum is called.

ichlfl is a flag for the inclusion/omission of chlorophyll fluorescence:

If *ichlfl* = 0, the is no chlorophyll fluorescence present.

If *ichlfl* = 1, chlorophyll fluorescence is present; routine chlz is called.

icdomfl is a flag for the inclusion/omission of CDOM fluorescence:

If *icdomfl* = 0, there is no CDOM fluorescence present.

If *icdomfl* = 1, CDOM fluorescence is present; routine acdom is called.

iraman is a flag for the inclusion/omission of Raman scattering:

If *iraman* = 0, there is no Raman scattering present.

If *iraman* = 1, Raman scattering is present.

RECORD 5.

This record defines the bottom boundary condition. It has the form

ibotm, rflbot

where

ibotm is a flag for the type of bottom boundary, as follows:

If *ibotm* = 0, the water column is infinitely deep. The water below depth $m = z_K = z(K)$, the last depth specified in Record 6 below, is taken to be homogeneous with IOP's equal to the values at depth *m*. The bi-directional radiance reflectance of the infinite layer of water below depth *m* is computed automatically from the assumed IOP's.

If *ibotm* = 1, the bottom is an opaque Lambertian reflecting surface located at depth *m*. The irradiance reflectance of the bottom is taken to be *rflbot, independent of wavelength.* Note that $0.0 \le rflbot \le 1.0$.

If *ibotm* = 2, the bottom is an opaque Lambertian reflecting surface located at depth *m*. The wavelength-dependent irradiance reflectance of the bottom is taken to be characteristic of *clean coral sand.*

If *ibotm* = 3, the bottom is an opaque Lambertian reflecting surface located at depth *m*. The wavelength-dependent irradiance reflectance of the bottom is taken to be characteristic of *green algae.*

If *ibotm* = 4, the bottom is an opaque Lambertian reflecting surface located at depth *m*. The wavelength-dependent irradiance reflectance of the bottom is taken to be characteristic of *brown algae.*

If *ibotm* = 5, the bottom is an opaque Lambertian reflecting surface located at depth *m*. The wavelength-dependent irradiance reflectance of the bottom is taken to be characteristic of *red algae.*

A value of *rflbot* is always read, but is used only if *ibotm* = 1.

RECORD 6.

This record first specifies whether the output "depth" refers to dimensionless optical depth ζ or geometric depth z, in meters. The record then defines the K depths at which output is desired. The last depth specified is taken to be the "maximum depth of interest," $m = \zeta_K$ or $m = z_K$. The water below the last depth is assumed to be homogeneous if *ibotm* = 0. If *ibotm* ≥ 1, the Lambertian bottom is placed at the last output depth. The record has the form

iop, K, d(1), *d*(2), ..., *d*(*K*)

where

iop is a flag for optical or geometric depth, as follows:

If *iop* = 0, then the *d* values are GEOMETRIC depths in meters.

If *iop* = 1, then the *d* values are OPTICAL depths.

K is the number of depths where output is desired.

d(1) = 0.0, *d*(2), ..., *d*(*K*) = *m* are the depths where output is desired.

Note that if the run contains inelastic scattering, then the run *must* use geometric depth. Note also that *d*(1) must always be 0.0, the depth in the water just below the mean air-water surface, and that *d*(*K*) is by definition the maximum depth of interest. Recall from the discussion of Section 4.1.4 that the default printout assumes that the output depths are specified as closely spaced PAIRS of depths, for example, 0.00, 0.01, 1.00, 1.01,...,10.00, 10.01.

RECORD 7.

This record gives information needed for specifying the sky radiance distribution. The general form of the record is

nsky, *skyspec*(1), *skyspec*(2), ..., *skyspec*(*nsky*)

where *nsky* is the number of values to be read in the remainder of the record. The exact form of the record depends on the version of qasky being used:

If qasky from file qacardsky.f is being used, then Record 7 has the form

5, *rsky, card, Eotot, thetas, phis*

where

rsky is the ratio of background-sky to total scalar irradiance, 0.0 ≤ *rsky* ≤ 1.0. *rsky* = 0.0 for a black sky (sun only) and *rsky* = 1.0 for a fully overcast sky (no sun visible).

card is the cardioidal parameter [*card* = *C* in *Light and Water* Eq. (4.50)]. *card* = 0.0 gives a uniform background sky and *card* = 2.0 gives a cardioidal sky.

Eotot is the total spectral scalar irradiance due to sun and background sky incident onto the sea surface; the units are W m^{-2} nm^{-1}.

thetas is the solar zenith angle in degrees; *thetas* = 0.0 for the sun at the zenith and *thetas* = 90.0 for the sun at the horizon.

phis is the solar azimuthal angle in degrees *relative to the wind direction*. *phis* = 0.0 is downwind and *phis* = 90.0 places the sun at a right angle to the wind.

If qasky from file qarealsky.f is being used, then Record 7 can have either of two forms. If the input gives the solar zenith angle, then Record 7 has the form

4, 1.0, *thetas, phis, cloud*

where

1.0 is a flag telling qasky that the zenith angle is being given explicitly.

thetas is the solar zenith angle, defined as above.

phis is the solar azimuthal angle, defined as above.

cloud is the cloud cover, 0.0 ≤ *cloud* ≤ 1.0. *cloud* = 0.0 for a clear sky and *cloud* = 1.0 for a solid overcast.

If the input gives the time and location from which the sun's zenith angle is to be computed, then Record 7 has the form

7, 2.0, *fjday, rlat, rlon, hour, phis, cloud*

where

2.0 is a flag telling qasky that the zenith angle must be computed.

fjday is the Julian day: *fjday* = 1.0 for January 1.

rlat is the latitude in degrees; positive for north and negative for south.

rlon is the longitude in degrees; positive for east and negative for west.

hour is Greenwich Mean Time in hours (Pacific Standard Time plus 8 hours). Minutes are expressed as a fraction, for example, 21.40 is 9:25 PM GMT.

phis is the azimuthal angle, defined as above.

cloud is the cloud cover, defined as above.

Other parameters (such as aerosol type, ozone concentration, and relative humidity) required for the sky radiance computations are set to default values in qasky, but could be read in here if desired. See the extensive comments in file qarealsky.f for details.

Subroutine readin2 sets a number of parameters to default values. For example, the default printout has no printout of the radiance distribution or of radiance K functions, and irradiances and irradiance K functions are printed out at every other depth, as described in Section 4.1.4. The *idbug* parameter, which controls the amount of printout, is set to 0 to give the minimum amount of printout. The parameters defining these defaults can be changed as described in readin2 to get more printout, or readin2 can be changed to read in these parameter values at run time in order to get greater flexibility in the specification of these parameters.

4.3 Required Routines

In addition to the core code, Part 2 requires the following routines from *Numerical Recipes, Second Edition*:

indexx	generates an index array *indx*(1:*n*) for a given array *arr*(1:*n*) such that *arr*(*indx*(*j*)) is in ascending order for $j = 1, 2, ..., n$.
ludcmp lubksb	ludcmp and lubksb together are used to invert matrices via an LU decomposition and back substitution.
odeint rkqs rkck	odeint, rkqs, and rkck together are used to solve systems of ordinary differential equations via a Runge-Kutta algorithm with adaptive stepsize control. The default versions of odeint, rkqs, and rkck (as distributed by *Numerical Recipes*) are dimensioned for a maximum of 50 equations, which is not adequate for HYDROLIGHT. Therefore, the dimension statements in these three routines must be changed as shown in the following segment of FORTRAN code, which is taken from odeint:

```
c     the default Numerical Recipes parameter statement is
c     PARAMETER (MAXSTP=10000,NMAX=50,KMAXX=200,TINY=1.e-30)
c     For HYDROLIGHT 3.0, this should be changed to
      parameter(mxmu=15, mxeqn=2*mxmu*mxmu + 2*mxmu)
      PARAMETER (MAXSTP=1000,NMAX=mxeqn,KMAXX=200,TINY=1.e-15)
```

The same change in parameter NMAX must be made in routines rkqs and rkck.

qromb polint trapzd	qromb, polint, and trapzd together are used to integrate functions via Romberg's method.
spline splint	spline and splint together are used to determine cubic spline fits to data and then to evaluate the interpolating functions.

See *Numerical Recipes, Second Edition* for detailed descriptions of these routines. The user can replace these routines with equivalent ones.

4.4 Example Output

The distributed HYDROLIGHT 3.0 code contains several examples that illustrate a few of the ways in which Part 2 can be run. The associated make, run, input, and output files all have names like Input2.ex1, Input2.ex2, and so on. The make and run files are specific to the UNIX operating system; users with other operating systems will have to replace these files with corresponding files of job control language for their computer. Users should be able to duplicate the example output files after installing the HYDROLIGHT code on their computers. The following discussion of the example runs presumes that the reader has obtained the HYDROLIGHT source code and can refer to the various ASCII files as needed in order to follow along with the discussion.

4.4.1 Example 1: Discretizing a Phase Function

The first example discretizes the particle phase function defined in file pfpart.f. This phase function is based on Petzold's data; see *Light and Water*, pages 109-111 for details. The make file Make2.ex1 is first invoked (on a UNIX system) via <make -f Make2.ex1> to compile the source code and create an executable file. The *Numerical Recipes* routines must be added to the HYDROLIGHT source code before compilation. Note in file Make2.ex1 that file pfpart.f is the phase function file being compiled. The versions of abscat, qasky, and so on, used here are irrelevant, since those routines are never called during a discretization run. File Run1.ex1 is then used to run the model in

discretization mode. Note in file Run1.ex1 that the file of surface information created and saved in the example run of Part 1 is linked to the run with the name rtspec. Part 2 always reads rtspec to get information on the quad partitioning, which is required for the phase function discretization. The input records of Section 4.2 now come from file Input2.ex1, which contains only three records, and the printout goes to Output2.ex1. The discretized phase function is always written to a binary file named phase1; this file is then saved with a descriptive name (in this case, pfpart.20x24) for later use. The above process must be repeated for all phase functions to be used in subsequent runs. In so doing, the make file must of course be changed so as to compile the desired phase function (instead of pfpart.f, as was used in the example) and to create a new executable file.

The time required to discretize a phase function is extremely dependent on the values of *nsubmu, nsubphi,* and *incfwd.* The run time is most sensitive to the value of *incfwd,* because large values of *incfwd* greatly increase the numbers of subquads used in the evaluation of ***Light and Water*** Eq. (8.13) for near-forward scattering angles. The example run, which used (*nsubmu, nsubphi, incfwd*) = (3, 4, 10) and the quad partition of Fig. 1(a), required 2710 seconds on the author's SPARCstation 2. Discretizing the same phase function with (*nsubmu, nsubphi, incfwd*) = (3, 4, 5) required only 165 seconds, but the accuracy checks on the computed quad-averaged phase function were not satisfactory (checksums that should be 1.00 were as large as 1.11; see *Light and Water*, page 386). On the other hand, discretizing the pure water phase function, defined in file pfpure.f, with (*nsubmu, nsubphi, incfwd*) = (2, 2, 1) and the same quad partitioning required less than one second and gave very accurate checksums (values between 0.9998 and 1.0001).

4.4.2 Example 2: An Idealized Radiative Transfer Simulation

Example 2 illustrates running HYDROLIGHT in a manner that is convenient for idealized radiative transfer studies. This example solves the monochromatic RTE for a homogeneous, infinitely deep water body. For pedagogic purposes, suppose that we wish to define the inherent optical properties of the water body by the phase function $\tilde{\beta}$ and the albedo of single scattering, $\omega_o = b/c$. Here $c = a + b$ is the beam attenuation coefficient. We shall use $\omega_o = 0.75$ along with a one-term Henyey-Greenstein scattering phase function with an asymmetry parameter of $g = 0.9$. This version of routine phasef is found on file pfothg.f; note that this file is the one named in Make2.ex2. This phase function was discretized just as in Example 1 but with (*nsubmu, nsubphi, incfwd*) = (3, 4, 5), which gave excellent accuracy for this phase function.

For convenience of reference, the file Input2.ex2, which contains the run-time input as described in Section 4.2, contains the following records:

```
Run for Part 2, Example 2, of the User's Guide
1
0 0 0 0 0
550.0 0.1 0.3
0 0.02
1 18 0.0 0.01 0.10 0.11 0.25 0.26 0.5 0.51 1.0 1.01 2.0 2.01
3.0 3.01 4.0 4.01 5.0 5.01
5 0.0 0.0 1.0 45.0 0.
```

Note that the water is being modeled as a "one-component" system (*ncomp* = 1 in record 2). Therefore, the run file Run2.ex2 links with only one previously discretized phase function (stored with the descriptive name pfothg.20x24) and gives it the name phase1. Likewise, the values of a and b (to be specified in Record 4) represent the total absorption and scattering coefficients. The run is being made at a single wavelength, and there are no inelastic-scattering or internal-source effects in this run (Record 3). Since the water body is to be homogeneous, the version of abscat found on file abconst.f is loaded in Make2.ex2. In order to get the desired value of $\omega_o = 0.75$, we can use $a = 0.1$ m^{-1} and $b = 0.3$ m^{-1}. These values are read in as *areset* and *breset* in Record 4. Any other combination of a and b that gives $b/(a + b) = 0.75$ could also be used – the solution of the RTE *in terms of the optical depth* will be the same in either case. (The same optical depth will correspond to different *geometric* depths for other allowed values of a and b, such as $a = 1.0$ m^{-1} and $b = 3.0$ m^{-1}.) The value of the wavelength read in at run time (here 550.0 nm) is irrelevant, because values of *areset* = 0.1 m^{-1} and *breset* = 0.3 m^{-1} are used to define the a and b values returned by *abscat*. (Recall that the option of using *areset* and *breset* to define a and b at run time is available only with the version of abscat found on file abconst.f; other versions of abscat use the wavelength to determine the appropriate a and b values.)

The bottom is infinitely deep, so the value of *rflbot* = 0.02 is read but never used in the run. Record 6 requests the output in terms of optical depth, which is the relevant measure of depth from the viewpoint of radiative transfer theory. (Note that it is optical depth that determines the run time for a given computation.) The present version of routine abscat works with either optical or geometric depth, because the water is homogeneous. Output is requested at 18 pairs of closely spaced optical depths, as explained in Section 4.1.4. Note that Record 6 requires two lines to list the 18 depths; any number of continuation lines can be used in specifying the free-format input records.

Make2.ex2 compiles the version of qasky found on file qacardsky.f, so Record 7 gives the input requested by that routine, as described in Section 4.2. The sun is placed at a zenith angle of 45° in the downwind direction; the sky is otherwise black. The input solar scalar irradiance incident onto the sea surface, E_{od}, is set to 1.0 W m^{-2} nm^{-1}. For a collimated sky radiance as used in this example, $E_{od} = E_{\perp}$, where $E_{\perp}$ is the irradiance on a surface perpendicular to the sun's direct beam. Finally, the water surface is level (zero wind speed), because the run file Run2.ex2 attaches a version of file rtspec that was created just as in the example of Part 1, except that $U = 0$ was specified.

The printout from this run is found on file Output2.ex2, which has been annotated for ease of understanding. At the end of Part 2, the complete digital output is always written to a file named radian; this binary file is saved with a descriptive name, here Radian.ex2, for later graphical or other analysis.

4.4.3 Example 3: A Simulation Using Measured IOP's as Input

The next example shows how measured values of IOP's can be used in running HYDROLIGHT. A file named Data.ex3 (distributed with the HYDROLIGHT code) contains a set of $a(z,\lambda)$ and $c(z,\lambda)$ values measured with a WETLabs ac-3 meter (a commercially available instrument manufactured by Western Environmental Technology Laboratories, Inc.). The a and c values were recorded at discrete depths as the instrument was slowly lowered through the water column; measurements were taken simultaneously in three wavelength bands centered at 456, 488, and 532 nm. In this particular data set (which is extracted from a much larger one), measurements were recorded at 122 depths between z = 0.832 m and z = 14.894 m. As seen in the second header record of the data file, the measurements were made in Monterey Bay, California, on 28 August 1993 at 1523 local daylight savings time. The exact latitude and longitude are 36° 54.83′ N and 121° 56.00′ W.

The version of abscat found on file abmeas.f reads and processes this data file. We first note that the ac-3 instrument is calibrated to remove the contributions of pure water to a and c. This is done because oceanographers usually are interested in the contributions of particulate and dissolved matter to the IOP's, since those contributions carry the information that connects the optical properties of the water with biological and physical processes that are occurring in the water. However, the light field responds to the *total* IOP's. For simplicity in this example, we will model the water as a "single-component" system. Therefore, abscat first adds the wavelength-dependent pure-water a

and c values to the ac-3 measurements to get total a and c values. (The routine also checks for duplicate measurements at the same depth, which sometimes occur in the data.) Because HYDROLIGHT requires values of a and b as input, the abscat routine then computes $b = c - a$ at each measurement depth. Because HYDROLIGHT requires a and b values at any depth, the routine then fits the available values of a and b with cubic splines, which are used to define values at any depth between 0.832 and 14.894 m. Values of a and b at depths between 0 and 0.832 m are set to the respective values at 0.832 m; values of a and b at depths greater that 14.894 m are set to the values at 14.894 m. Routine abscat can therefore return total a and b values at any depth z, at each of the three wavelengths. This particular routine does no interpolation in wavelength because, for this example, we are going to run HYDROLIGHT only at one of the wavelengths given in the data. Figure 2 shows the measured a and c values (including the pure water contribution) and the corresponding b and ω_o values, at a wavelength of λ = 456 nm. (This plot was generated using the IDL routine mpac3.pro, which is distributed along with the HYDROLIGHT code.)

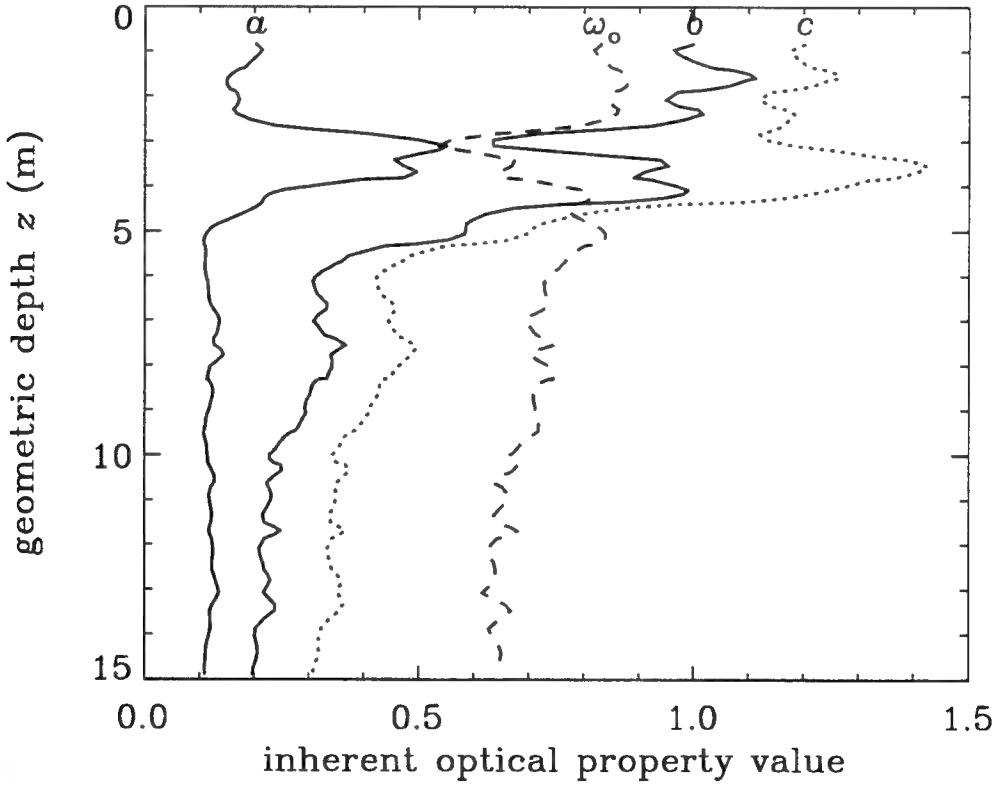

Fig. 2. Inherent optical properties at λ = 456 nm, as used in Example 3. a, b, and c have units of m^{-1}, and ω_o is dimensionless.

The phase function was not measured in the Monterey Bay experiment under consideration. Therefore, in this example, we shall assume that the total phase function of the water is adequately described by the typical particle phase function discretized in Example 1. This is a reasonable approximation since the scattering by the water itself is negligible compared to the scattering by particles in this rather turbid coastal water.

Figure 2 shows that the IOP's have a lot of "structure" or variability near the surface, especially between two and six meters depth. Below six meters the water is much more uniform. In running HYDROLIGHT we therefore request output at many depths near the surface, in order to resolve the IOP effects on the light field. As seen in file Input2.ex3, output is requested at closely spaced pairs of depths every 0.5 m between the surface and 2 m depth, every 0.25 m between 2 and 6 m depth, and every meter below 6 m depth. A bottom boundary condition for homogeneous, infinitely deep water is applied at depth m = 15 m.

A 10 nm wide wavelength band centered on the nominal ac-3 wavelength of 456 nm was used in the simulation. Thus Input2.ex3 shows *nwave* = 1, *waveb*(1) = 451.0, and *waveb*(2) = 461.0, as described in Section 4.2, Record 4. This corresponds roughly to the bandwidth of various radiometric instruments that were used to collect data simultaneously with the ac-3 data. In a complete analysis of the experiment, we would of course wish to compare the HYDROLIGHT output with values measured by such instruments.

The "real sky" version of qasky is used for this simulation. The last line of Input2.ex3 gives qasky the Julian day, Greenwich mean time, latitude and longitude for the measured data, as described in Section 4.2, Record 7. A clear sky with default atmospheric parameters was used. The values of these atmospheric parameters are given in the printout. The wind speed was not measured, so the simulation was performed with U = 5 m s^{-1} as a reasonable guess; the appropriate file from Part 1 is linked with the run in file Run2.ex3.

Figure 3 shows the upwelling (E_u) and downwelling (E_d) plane irradiances and the "nadir-viewing," or upwelling, radiance L_u (i.e., radiance traveling toward the zenith) for this simulation. Note that since we have used a realistic sky, these are absolute values that can be compared directly with measured values. (Note however, that the assumed sea state and atmospheric conditions may differ considerably from those during the actual experiment. Any such errors in the input to HYDROLIGHT of course induce corresponding errors in the radiometric quantities predicted by HYDROLIGHT.) The irradiances have units of W m^{-2} nm^{-1} and L_u has units of W m^{-2} sr^{-1} nm^{-1}; that is, these are

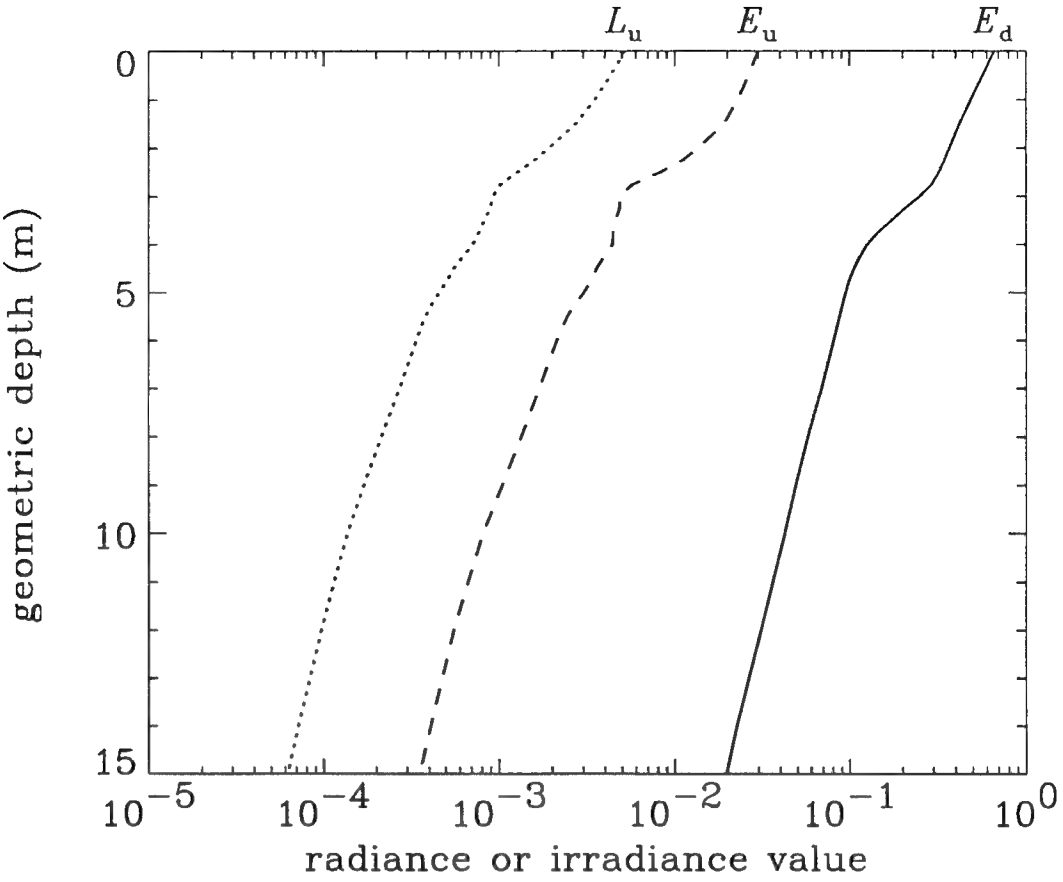

Fig. 3. Upwelling and downwelling plane irradiances and upwelling radiance at λ = 456 nm, as computed by HYDROLIGHT 3.0 for Example 3. E_u and E_d have units of W m^{-2} nm^{-1} and L_u has units of W m^{-2} sr^{-1} nm^{-1}.

band-averaged values, not band-integrated values, in accordance with Eq. (1). (This figure was generated with IDL routine mpirrad.pro, which is designed to read the file of digital output written by Part 2. Users with IDL can use this program as a template to generate IDL programs for plotting other quantities of interest.)

The irradiances and nadir-viewing radiance plotted in Fig. 3 show a clear effect of the structure in the IOP's on the radiometric quantities. However, plots using log-linear axes, as is customary for such quantities, do not show the wealth of information that is contained in the digital data generated by HYDROLIGHT. Figure 4 shows K_d as computed from the closely spaced pairs of digital E_d data, as described in Section 4.1.4. The figure also shows the irradiance reflectance $R = E_u/E_d$ (times 10 for clarity in the plot) and the mean cosine μ of the entire radiance distribution. This figure shows that these apparent

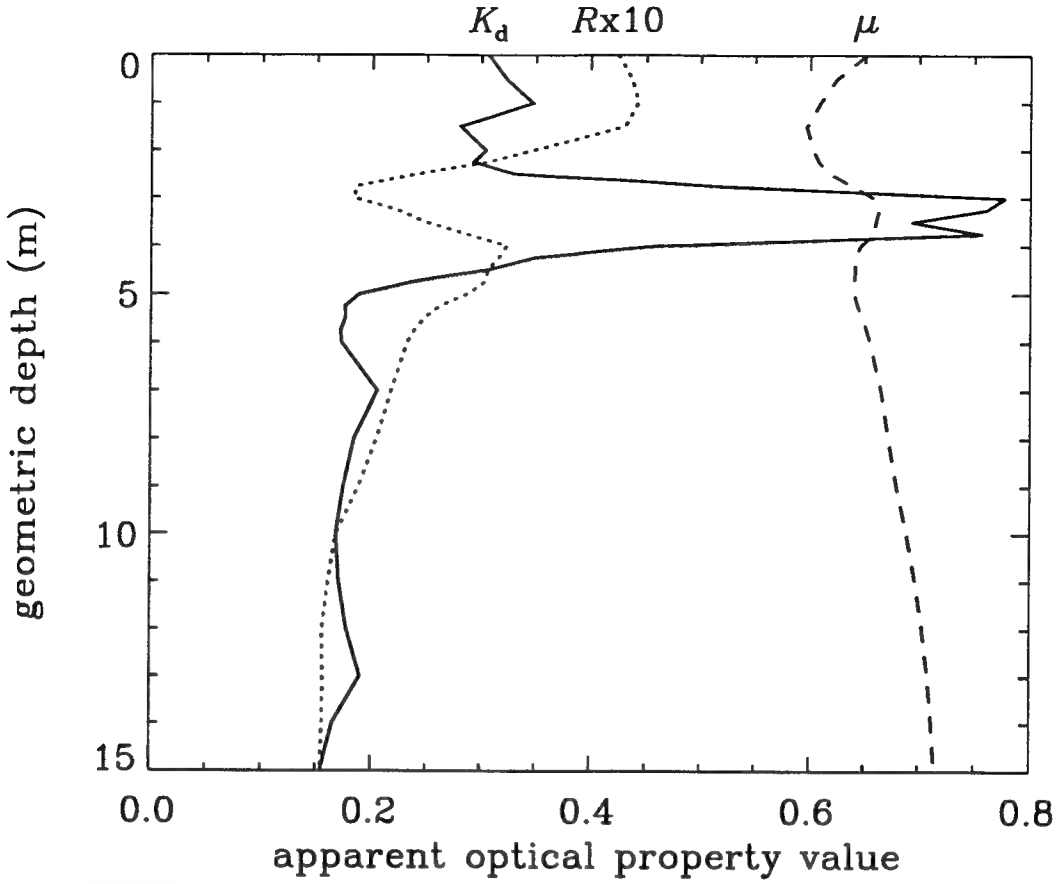

Figure 4. Apparent optical properties at λ = 456 nm, as computed for Example 3. K_d has units of m^{-1}; R and μ are dimensionless. This figure shows the computed values when output is obtained with high depth resolution.

optical properties contain a great deal of information about the vertical structure of the water column. Note, for example, how well the vertical structure of K_d correlates with the absorption coefficient a as shown in Figure 2, and how well R correlates with the scattering coefficient b of Fig. 2. This behavior is exactly as expected, since it is well known that K_d is an "absorption-like" AOP, and since R depends on how much of the downwelling light is scattered into upward directions. (Figure 4 was generated using IDL routine mpaop.pro.)

However, such detail in the vertical structure of the HYDROLIGHT output – or of measured values – is seen only if output is requested at a sufficient number of depths. Figure 5 shows the AOP's obtained in a repeat of the above simulation with all of the physical parameters being exactly the same, but with output requested only at two-meter

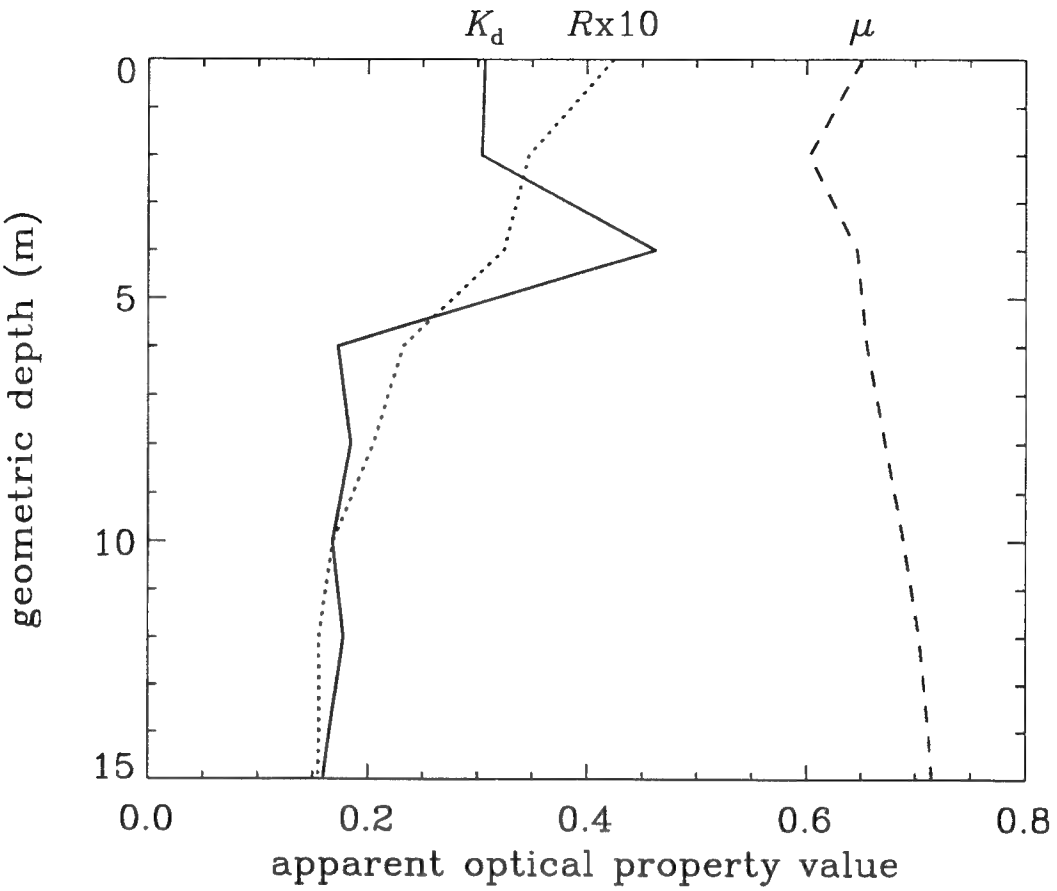

Figure 5. Apparent optical properties at λ = 456 nm, as computed for Example 3. This figure shows the computed values when output is obtained with low depth resolution. The curves should be compared with those of Figure 4.

intervals, that is, at z = 0.00, 0.01, 2.00, 2.01, 4.00, 4.01, ... In both cases HYDROLIGHT solves exactly the same physical problem. Thus the values shown in Figu.5 are the same as the values shown in Fig. 4 at the corresponding depths of 0, 2, 4, ... However, the detailed vertical structure seen in Fig. 4 has been lost in Fig. 5, because not enough values were saved for plotting.

Although the physical problem solved in generating Figs. 4 and 5 is the same, the run with the greater amount of depth information did require longer to run, because the core code had to work with larger internal arrays (whose sizes depend on the number of depths where results are to be saved).

4.4.4 Example 4: A Simulation of Case 2 Water, Including Inelastic Scattering

As a final example of running HYDROLIGHT, we simulate the light field in shallow Case 2 water. The water is modeled as a three component system: pure water plus chlorophyll-bearing particles plus colored dissolved organic matter. The run includes Raman scattering by the water itself and fluorescence both by chlorophyll and CDOM. The bottom is placed at a depth of 8 m and is modeled as being covered with green algae. The wavelength bands are chosen to approximate the bandwidths typically used by hyperspectral remote sensing instruments.

Routine chlz defines a chlorophyll profile that depends on depth. The chlorophyll concentration for this run increases from roughly 1 mg m^{-3} at the surface to 4.8 mg m^{-3} at z = 5 m, and then decreases to 2.1 mg m^{-3} at 8 m. Routine acdom defines CDOM absorption that depends on depth and wavelength. At 440 nm, the absorption by CDOM is 0.1 m^{-1} at the surface; this value decreases to 0.02 m^{-1} at z = 8 m. The CDOM absorption decreases exponentially with wavelength; it is negligible at red wavelengths, but is triple the above values at 350 nm.

The version of routine abscat found on file abcase2.f calls routines chlz and acdom during the computation of the absorption and scattering coefficients at various depths and wavelengths. Standard bio-optical models (as documented in file abcase2.f) are used to convert the chlorophyll concentration to a and b values for the particle component (although those models were developed for use in Case 1 waters only). Depending on the depth and wavelength, the total absorption can be dominated by the water, by chlorophyll, or by CDOM. Scattering by chlorophyll-bearing particles is always much greater than scattering by the water; the CDOM is assumed to be nonscattering. This model for a and b gives albedos of single scattering, ω_o, that range from less that 0.3 (at 700 nm, where absorption by the water is very high) to more than 0.85 (near 550 nm, where absorption is lowest). The geometric depth of 8 m corresponds to over 10 optical depths at 350 nm (where both CDOM absorption and particle scattering are high) and to less that 6 optical depths near 550 nm. Figure 6 shows the IOP's a, b, c, and ω_o as functions of depth and wavelength. (This figure was generated using IDL routine mpasurf.pro.)

The discretized phase functions for pure water and for particles were computed as in Example 1, using the phasef routines from files pfpure.f and pfpart.f, respectively.

The simulation includes all wavelengths from 350 to 700 nm, as is necessary when fluorescence effects are included. Wavelength bands are 10 nm wide between 350 and 650

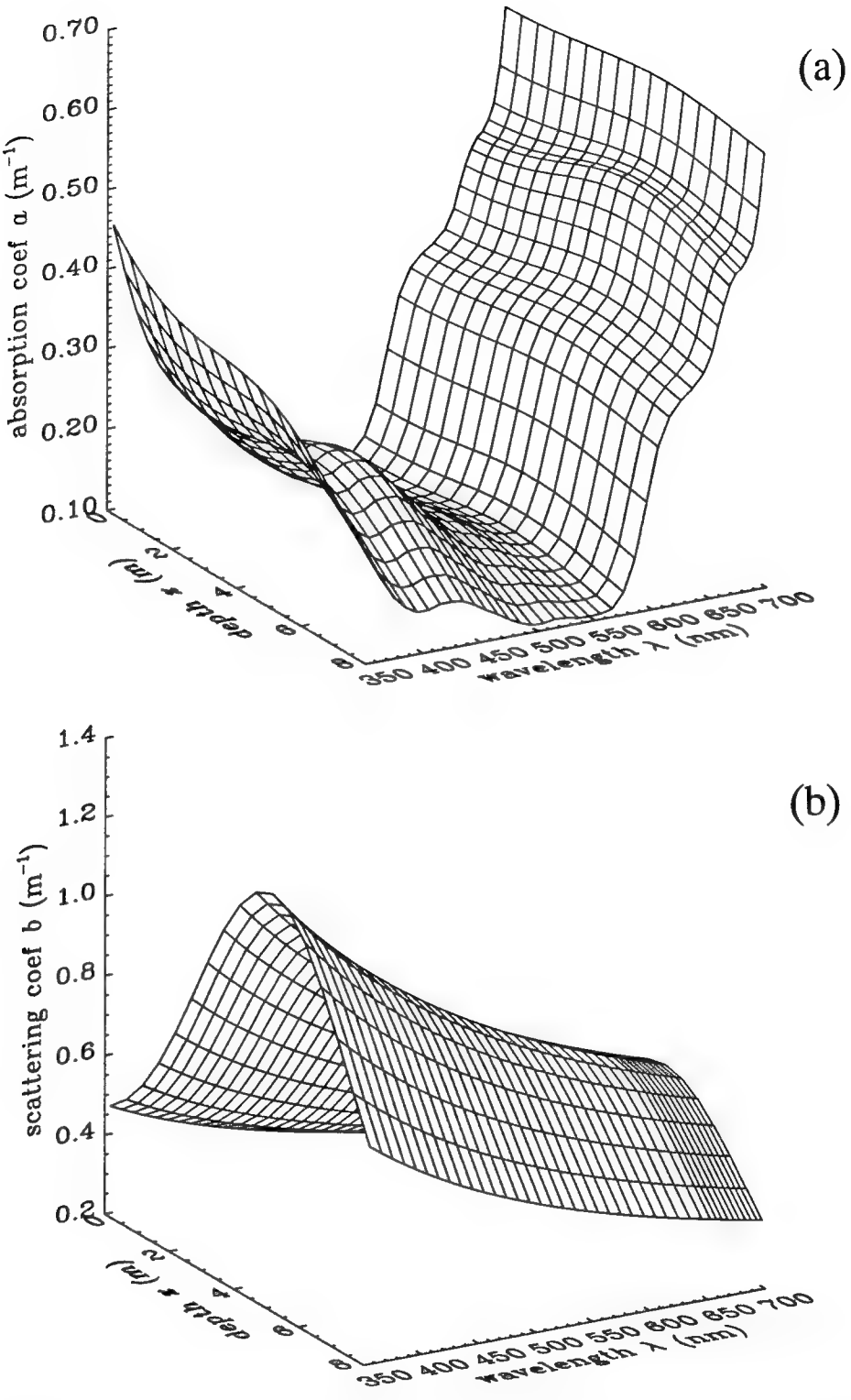

Fig. 6. Inherent optical properties as function of depth and wavelength, as used in Example 4. Panel (a) is the absorption coefficient *a*, panel (b) is the scattering coefficient *b*, panel (c) is the beam attenuation coefficient *c*, and panel (d) is the albedo of single scattering ω_o.

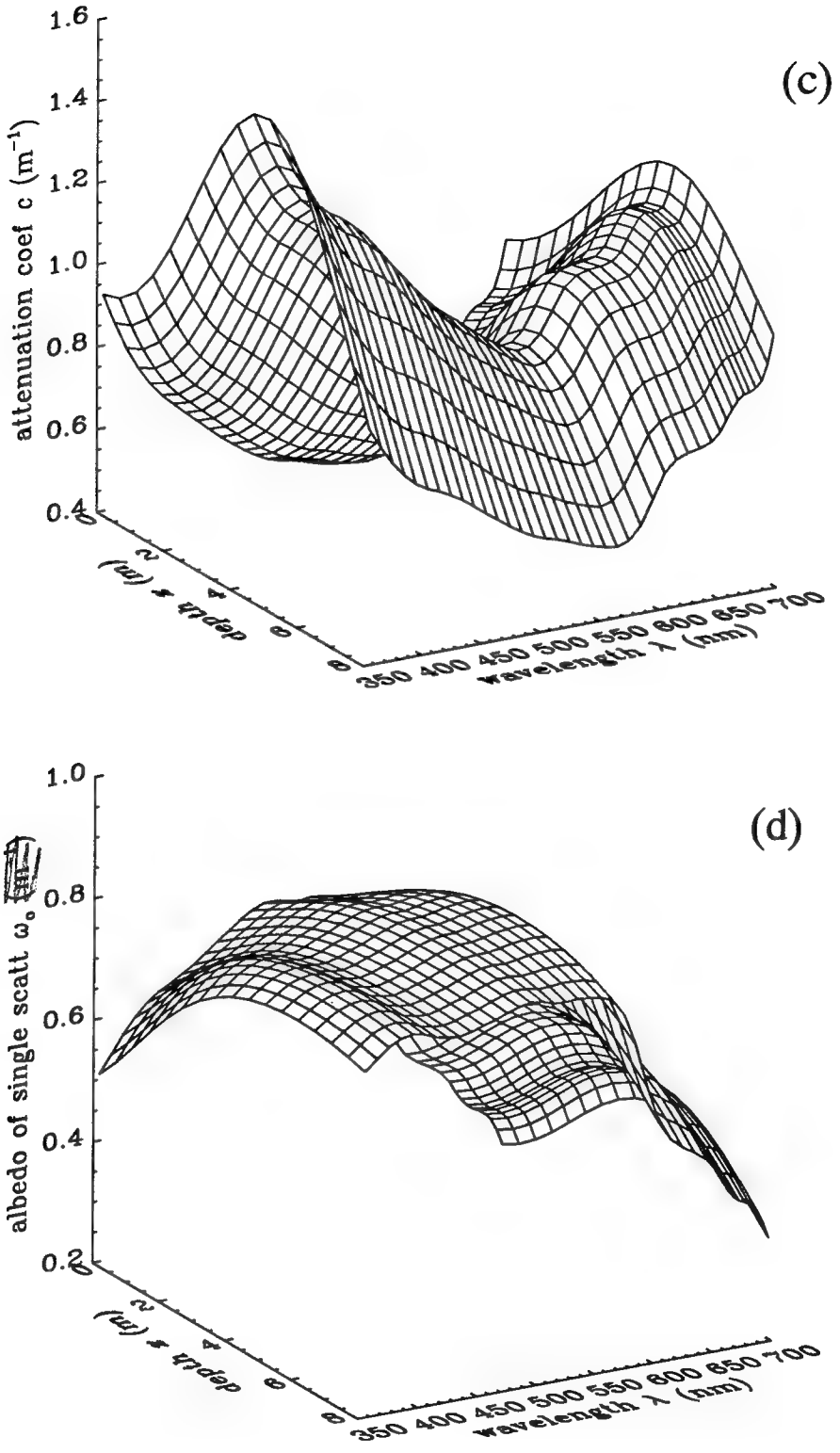

Fig. 6 continued.

nm and 5 nm wide between 650 and 700 nm, for a total of 40 bands. The 5 nm bands were chosen to get greater wavelength resolution of the chlorophyll fluorescence band, which is centered near 685 nm.

Output was requested at closely spaced pairs of depths every half meter between the surface and the bottom, for a total of 34 depths. The bottom was modeled as a Lambertian surface with a wavelength-dependent irradiance reflectance that is based on measurements of a green algae.

The sun is placed at a zenith angle of 60° and an azimuthal angle of 45° (relative to the downwind direction). The sky is assumed to have a cloud cover of *cloud* = 0.7. This number is probably best thought of as representing a uniformly overcast sky through which the sun is still visible. [Technically, *cloud* is the "total cloud amount" in the equations of Kasten and Czeplak (1980).] The wind speed is taken to be 5 m s^{-1}.

It is very important to note in run file Run2.ex4 that the file pfpure.20x24, which contains the discretized phase function for pure water for the quad partition discussed in the example of Part 1, is linked to this run as local file phase1. Likewise, file pfpart.20x24, which contains the discretized particle phase function, is linked to local file phase2. This linkage is determined by the choice of component 1 being pure water and component 2 being particles in routine abscat. Because component 3 in abscat is CDOM, and because CDOM is taken to be nonscattering (i.e., abscat always sets $b_3 = 0$ for component 3), it is immaterial what phase function is linked to the name phase3: the numerical values read from phase3 will always be multiplied by $b_3 = 0$ when computing the total phase function. However, the code requires a phase function for each component, since it does not know *a priori* that component 3 will have a zero scattering coefficient. In the present case, the phase function for pure water is also linked as phase3.

This run required about 5.5 hours on the author's SPARCstation 2 workstation. It should be noted that the simulation requires the solution of the RTE for 40 wavelength bands, with three types of inelastic scattering connecting the bands. On average, each wavelength solution was taken to about eight optical depths. Thus the run time corresponds to about one minute per wavelength per optical depth. The archival output file Output2.ex4 is now much larger than before (over 600 Kbytes), since the same information is provided for each wavelength. Likewise, the binary file of radiance data (radian, which is saved with the descriptive name Radian.ex4) is now about 4.4 Mbytes.

The next figures very briefly illustrate the kind of graphical output that can be obtained from the digital file radian, which is always created at the end of Part 2.

Figure 7 shows depth profiles of the downwelling and upwelling plane irradiances for two wavelength bands. Values for the band from 550 to 560 nm are denoted by "555" in the figure, and the band from 650 to 655 nm is denoted by "653." As we might anticipate, the downwelling irradiance E_d decays much quicker with depth at 653 than at 555, because of the high absorption by water at 653. The behavior of E_u is more interesting. At 555, where the total absorption and total attenuation are lowest, E_u is nearly constant with depth. This is a consequence of the high bottom reflectance (0.27, as seen in the printout) and the relative clarity of the water. At 653, the bottom reflectance (now 0.13) noticeably influences the E_u profile only in the two or three meters just above the bottom. (This plot was generated using the IDL routine found on file mpirrad2.pro.)

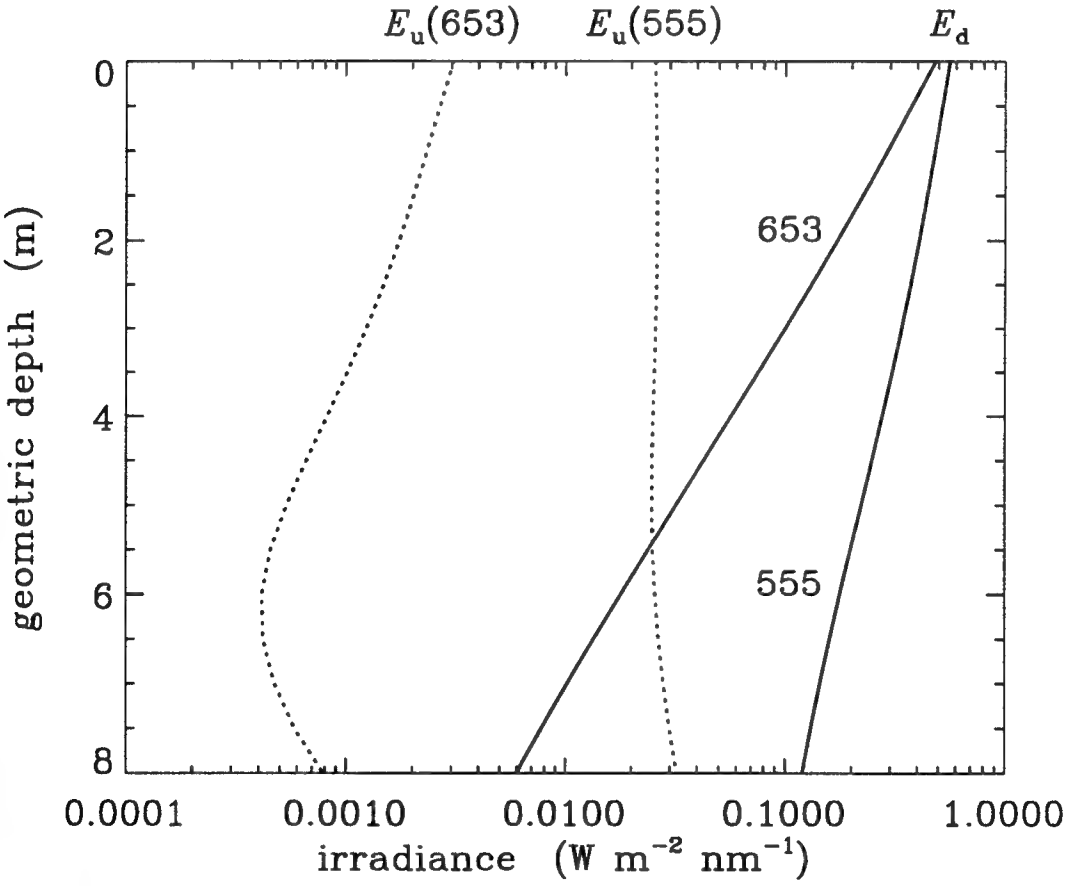

Figure 7. Irradiances for two wavelength bands, as computed for Example 4. Band 550-560 nm is labeled "555" and band 650-655 nm is labeled "653."

Figure 8 shows K_d and K_u, the diffuse attenuation functions for E_d and E_u, as functions of wavelength for a depth of z = 1 m. [These K values were computed using Eq. (2) and the E values at z = 1.00 and 1.01 m.] The curve for K_d is similar to that for total absorption [not shown, but recall Fig. 6(a)]. The curve for K_u shows both bottom and fluorescence effects. K_u is slightly negative for wavelengths near 550 nm, which corresponds to the slight increase of E_u with depth seen in Fig. 7 at z = 1 m. The pronounced dip in the K_u curve near 685 nm results from chlorophyll fluorescence. The vertical lines at the bottom of the figure show the wavelength bands used in this simulation. (This plot was generated using the IDL routine found on file mpKwavez.pro.)

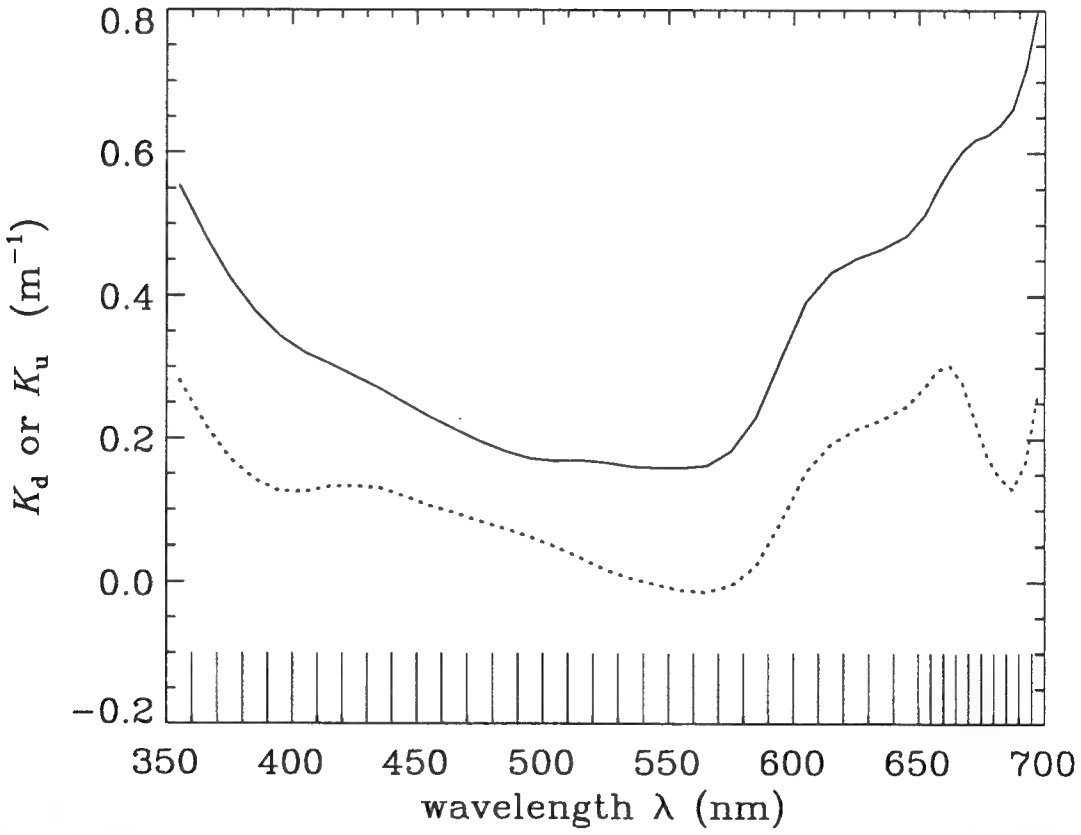

Figure 8. Diffuse attenuation functions K_d (solid line) and K_u (dotted line) at a depth of z = 1 m, as computed for Example 4. The vertical lines at the bottom show the wavelength bands used in the simulation.

Figure 9 shows the remote sensing reflectance $R_{rs} = L_w / E_d$ (where L_w is the water-leaving radiance, i.e., the total radiance minus the reflected sky radiance; and E_d is evaluated just above the water surface). The figure shows the results for two independent simulations: the one under discussion, which has a green-algae bottom, and an otherwise identical simulation, which used a sandy bottom (*ibotm* = 2 in Record 5 of the input). This figure shows how the bottom type can influence a remotely sensed signal even though the water is optically rather deep. As expected, the bottom effect is greatest in the green part of the spectrum, where the water is most transparent and where scattering dominates absorption: the high scattering allows sunlight to reach the bottom, be reflected, and return to the surface, carrying information about the bottom with it. Although the bottom albedos differ by a factor of three to almost eight between 600 and 700 nm, the different bottoms

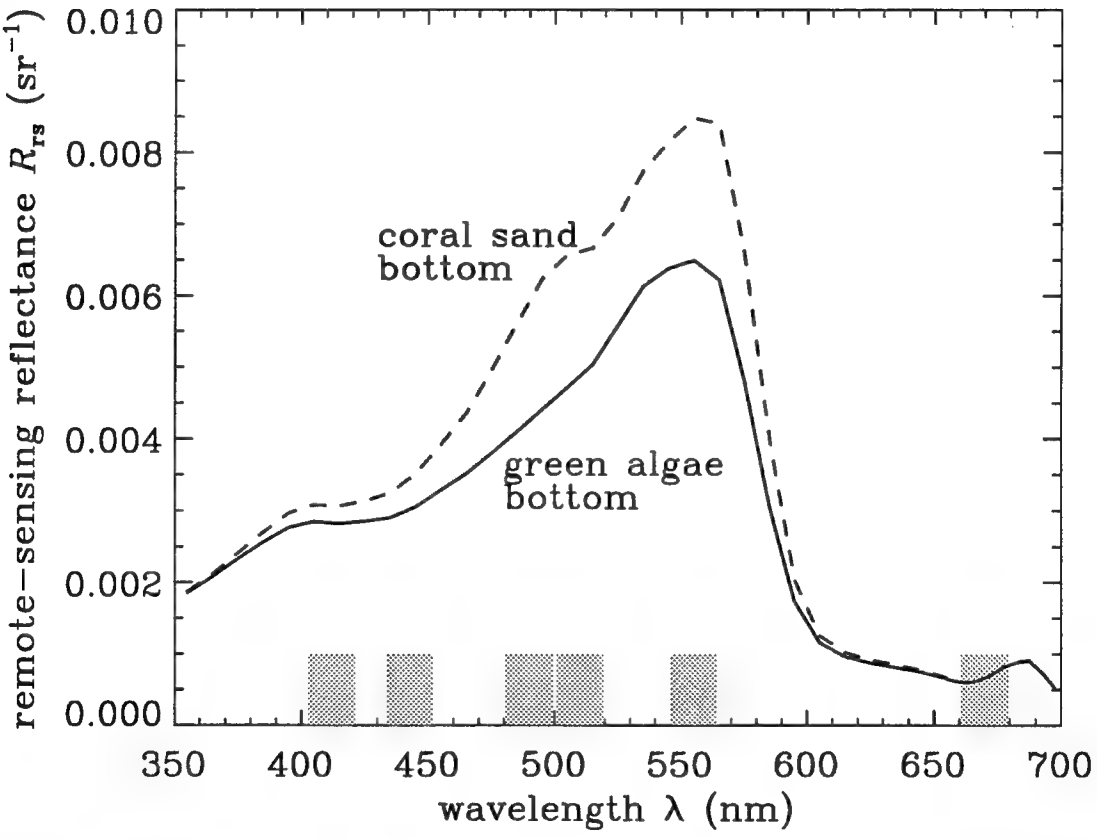

Figure 9. The remote-sensing reflectance R_{rs} as computed for Example 4 (solid line) and as computed in an identical simulation, but the a different bottom type (dashed line). The shaded bands at the bottom show the SeaWiFS sensor bands.

scarcely affect R_{rs} there, because of the high absorption by the water itself at those wavelengths. (It should be noted from the printout of K_d that 8 m is less than two diffuse attenuation depths at green wavelengths, although the optical depth as computed from the beam attenuation is about six. At red wavelengths, 8 m corresponds to four or more diffuse attenuation depths.) The shaded bands at the bottom of the figure show the sensor bands of the SeaWiFS ocean color satellite. (This figure was generated with IDL routine mpRrs.pro.)

Figure 10 shows the radiance distribution as a function of direction for three depths, z = 1, 5, and 7 m, and for the wavelength band from 450 to 460 nm. The plot shows a "slice" of the radiance distribution taken in the ϕ plane of the sun's rays; recall that the sun was placed at an azimuthal angle of ϕ = 45°. The "viewing direction" in the figure is the

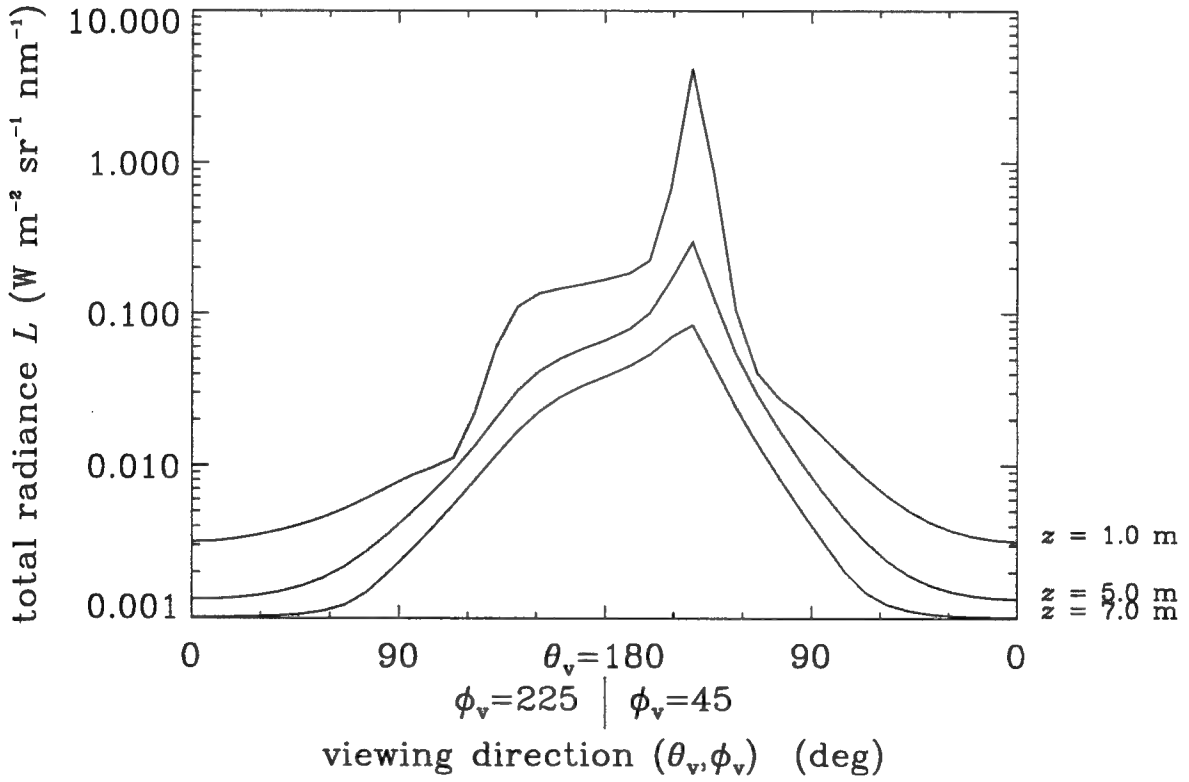

Figure 10. The radiance distribution in the plane of the sun's rays for the 450-460 nm band and at three different depths, as computed for Example 4. The maximum occurs when looking toward the sun's refracted rays. Viewing direction θ_v = 0 is looking straight down; θ_v = 180° is looking straight up.

direction a sensor would point in order to detect photons traveling in the opposite direction. Since HYDROLIGHT measures the polar angle from 0 at the nadir, a viewing direction of $\theta_v = 180°$ is looking straight up, seeing the radiance heading straight down; $\theta_v = 0$ corresponds to looking straight down and seeing the upwelling radiance L_u. The radiance maxima at $\theta_v \approx 40°$ and $\phi_v = 45°$ (i.e., looking toward the sun; $\phi_v = 225°$ is looking away from the sun) are due to the sun's direct rays after refraction at the air-water surface. These maxima become less prominent with depth as scattering makes the radiance distribution more diffuse. (Figure 10 was generated with IDL routine mprad.pro.)

The radian file of course contains the complete radiance distribution at all depths, wavelengths, and directions. Figure 11 shows the total radiance at $z = 5$ m in the ϕ plane of the sun, as a function of the polar viewing direction θ_v and wavelength. The 5 m curve in Fig. 10 corresponds to a slice through the surface at $\lambda = 455$ nm in Fig. 11. Many figures of this and similar types can be generated from the data on the radian file. (This figure was generated with IDL routine mpradsurf.pro.)

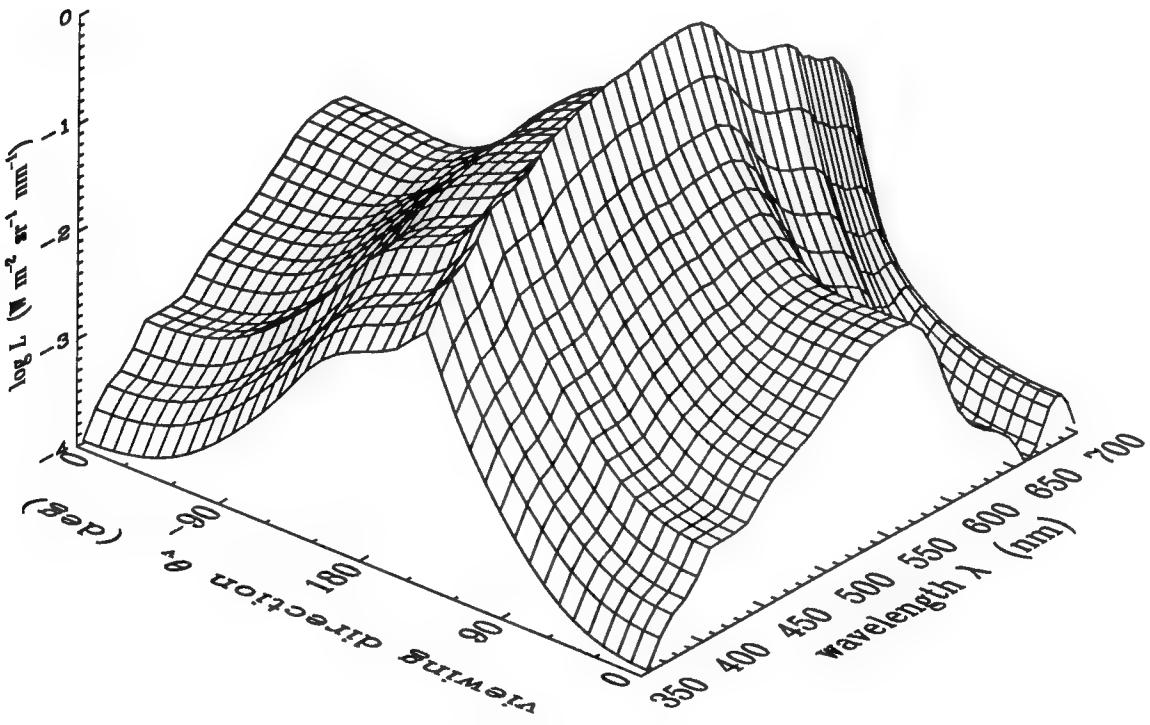

Fig. 11. The radiance distribution in the plane of the sun's rays, as a function of viewing direction and wavelength, at a depth of $z = 5$ m. The 5 m curve in Figure 10 is a cut through this surface at $\lambda = 455$ nm.

5. ADVICE ON USING HYDROLIGHT 3.0

It is worthwhile to summarize some of the previous comments about computer run times and to add some caveats about the numerical algorithms and the HYDROLIGHT model in general.

5.1 Comments on Running HYDROLIGHT 3.0 Efficiently

When contemplating a long series of runs, it is often a good idea to make a few trial runs of HYDROLIGHT, or at least to make a back-of-the-envelope calculation, in order to get a quantitative estimate of certain parameters. Perhaps the most important such parameter is the dimensionless optical depth ζ, which in homogeneous water is just the beam attenuation coefficient c times the geometric depth z. Optical depth is the fundamental measure of depth in radiative transfer theory, and *it is the optical depth that determines how long it takes to solve for the radiance distribution down to the maximum depth of interest, m.* Consider, for example, two homogeneous water bodies: the first has $c = 0.1$ m^{-1} and the second has $c = 1.0$ m^{-1}. Running HYDROLIGHT to a geometric depth of $m = 50$ m in the first water body will require the same amount of computer time as running the model to a depth of $m = 5$ m in the second water body, because both geometric depths correspond to 5 optical depths.

This optical depth influence on the run time can lead to large computation times in certain situations. Consider, for example, a simulation of clear, open-ocean water. In such water, light penetrates to great depths at blue wavelengths. In a study of biological productivity it might therefore be necessary to find the radiance to a depth of 100 m or more for blue wavelengths in order to compute the scalar irradiance throughout the euphotic zone, as is necessary for biological studies. At blue wavelengths, 100 m might correspond to only a few optical depths, and the solution will be rapid. However, at red wavelengths, where absorption by the water itself is of order 0.5 m^{-1}, 100 m corresponds to 50 or more optical depths. The solution of the RTE to 100 m at red wavelengths will therefore require a very long time. (In addition, solutions to such extreme optical depths may cause numerical problems on some computers because the radiance at 50 optical depths will be of order $e^{-50} \sim 10^{-22}$ of its surface value.) Moreover, the solution to such depths is unnecessary at red wavelengths, because the euphotic zone is much shallower as a consequence of the high absorption.

One way around this problem (for the clear-water example under discussion) is to make one run of HYDROLIGHT to 100 m for the wavelengths from 400 to 500 nm, say, then make another run to a lesser depth for wavelengths from 500 to 600 nm, and then make a third run (perhaps to only 10 m) for the 600-700 nm region. The outputs from the three runs can then be combined to obtain the lightfield throughout the euphotic zone for all visible wavelengths. This way around the optical depth problem is inelegant, but workable for simulations that do *not* include inelastic scattering effects. If inelastic scattering is included in the run, then all wavelengths must be run simultaneously so that a given wavelength can receive the inelastically scattered light from all shorter wavelengths. However, in clear waters, where the optical depth problem arises, fluorescence effects are likely to be minimal and probably can be omitted from the run without significant error. Raman scattering may, however, be significant in clear water, for certain applications.

Note that this "optical depth problem" did not occur in Example 4 above because the turbid, Case 2 water had rather high attenuation at all wavelengths, and thus 8 m corresponded to roughly the same optical depth (6 to 10) at all wavelengths. It was thus reasonable to make one run for all wavelengths. Time has not permitted the investigation of ways for the HYDROLIGHT algorithms to "dynamically" determine the solution depth necessary at each given wavelength.

For remote sensing applications, it is often sufficient to run HYDROLIGHT only to two or three "diffuse attenuation depths," where the diffuse attenuation depth at a given wavelength is $1/K_d$. This number is usually computed using the value of K_d just below the sea surface, even though K_d varies somewhat with depth even in homogeneous waters. Although bio-optical models exist for K_d in Case 1 waters (see, for example, Section 3.10 of *Light and Water*), it is usually expedient to make a few test runs of HYDROLIGHT at blue or green wavelengths to get an accurate estimate of the diffuse attenuation depth for any water type or environmental condition. The model can then be run to the corresponding maximum geometric depth m.

Running HYDROLIGHT only to a few diffuse attenuation depths in a remote-sensing study and running it to different geometric depths at different wavelengths as in the clear-water example above illustrate a general rule for efficient use of the model: *do not do more computation than is necessary to solve the problem at hand.* As another example of this rule, one should omit inelastic scattering effects if they are of negligible importance in a particular problem. Note, however, that for a given water body, fluorescence might be an important part of the signal in a remote sensing study and at the

same time be totally unimportant in the computation of water heating. The importance of inelastic scattering in a particular problem can be quantified by making occasional duplicate runs with and without inelastic scattering.

The run time for Part 1 is determined largely by the total number of rays being traced in the Monte Carlo estimation of the four radiance transfer functions for the air-water surface. Since these are one-time calculations for a given wind speed and quad partition, it usually makes sense to err on the side of tracing more rays than necessary to get acceptable statistical estimates. When in doubt about the accuracy of the estimates, make duplicate runs as described in Section 3.1.

The run time for Part 2 is determined largely by the optical depth, as discussed above, and by the total number of quads *squared*. (Recall, for example, how the phase function discretization must compute the quad-averaged phase function for all pairs of quads.) Therefore, running the model to the minimum necessary depth and using a quad partition with the minimum acceptable angular resolution can reduce computation times by factors of ten or more, compared to using deeper depths and more quads than are really necessary for the solution of a given problem. The next example illustrates the dependence of the run time on the quad resolution.

Many users of HYDROLIGHT 3.0 are interested only in the various irradiances, which are computed from the solution radiance distribution. Because the irradiances are computed from integrals of the radiance over direction, they are in principle independent of the quad partitioning. However, there is usually some difference in the irradiances computed from otherwise identical runs having different quad partitions because of the different "smearing out" of the sun's actual position. To illustrate this effect, two runs were made with the quad partitions shown in Figs. 1(a) and 1(d). The sun was placed at a nominal zenith angle of 45° in an otherwise black sky. In the run with the 20×24 quad partition of Fig. 1(a), the sun fell in a quad that stretched from $\theta \approx 43°$ to $\theta \approx 52°$. In the run with the 30×36 quad partition of Fig. 1(d), the sun fell in a quad that stretched from about 41° to 47°. These differences in the position of the quad containing the input solar radiance led to a difference of about seven percent in the values of E_d incident onto the sea surface, because of the different average values of $\cos\theta$ over the two solar quads. This difference was then carried though the calculations and even became slightly greater with depth because of the slightly different directions of the sun's refracted beam underwater. However, this was a rather extreme case, since the water surface was level and the background sky was black. When the same comparison was made with a cardioidal sky (i.e., for a heavy overcast with no discernable sun), the E_d values for the two quad

partitions agreed to better than 0.2 percent, and all other quantities agreed to within a fraction of a percent.

In the black-sky simulations just discussed, the 30×36 quad partition required 1033 seconds to run Part 2 to five optical depths, but the 20×24 partition required only 151 seconds. If errors in the computed irradiances of order 10 percent are acceptable, then it makes no sense to use a quad partition like that shown in Fig. 1(d), when the quad partition shown in Fig. 1(a) will give acceptably accurate irradiances. If irradiances or other quantities really are required to have an accuracy of better than a few percent, or if it is necessary to resolve directional changes in the radiance distribution on the scale of a few angular degrees, then the use of a fine quad partition like that of Fig. 1(d) may be justified.

5.2 Comments on the Input

In any case, HYDROLIGHT 3.0 is only as good as its input. If high absolute accuracy is required in the HYDROLIGHT output, then the various user-supplied inputs to the model must be of comparable accuracy. In particular, the incident sky radiance, the absorption and scattering coefficients, and the shape of the phase function all must be of great accuracy if the model output based upon those values is to be held to a high standard. It is seldom the case that such quantities are measured to accuracies of even a few percent. Commonly used bio-optical models for a and b, as are found in the **abscat** routines on files **abcase1.f** and **abcase2.f**, are often in error by a factor of two or more for a given chlorophyll concentration. We therefore should not be surprised if modeled and measured irradiances differ greatly in a situation where the chlorophyll concentration was measured and then converted into a and b values for use in HYDROLIGHT. Likewise, significant errors can occur in the computation of irradiances if the sky radiance and irradiance were modeled (i.e., guessed) rather than measured. Of course, some apparent optical properties are less affected by such errors, which is part of their virtue. K functions, for example, are completely insensitive to errors in the magnitude of the input sky irradiance, although they are still affected by errors in the IOP's.

5.3 Comments on the Physical Model

The limitations of the underlying HYDROLIGHT physical model also must be kept in mind. One of the most important of these limitations concerns the air-water surface. HYDROLIGHT 3.0 models the wind-blown air-water surface using the Cox-Munk capillary

wave-slope statistics. This treatment is adequate for wind speeds less than roughly 10 m s^{-1} and for lines of sight that are roughly 20° or more away from the horizon. At higher wind speeds, foam on the sea-surface may significantly change the reflectance properties of the surface (although there is some debate as to how important foam effects are). Also, the overall slope statistics of the sea surface are somewhat different in the presence of gravity waves, which are inevitably present at wind speeds of more than a few meters per second. For nearly horizontal lines of sight (e.g., near sunrise or sunset, or when we are interested in the radiance leaving the water in an almost horizontal direction regardless of solar elevation), wave shadowing by gravity waves can significantly modify the radiance transfer properties of the surface, compared to the predictions of the capillary-wave treatment. Clearly, any radiometric effects attributable to gravity waves will be absent in the HYDROLIGHT 3.0 predictions.

Small gravity waves also give rise to pronounced focusing of the sun's direct rays in the upper few meters of the water. These wave-focusing fluctuations may or may not be resolved by a particular sensor. Thus measured and modeled values of E_d, say, might differ near the water surface, either because such time-dependent effects are omitted from the HYDROLIGHT model or because the sensor did not record a true time average of the rapidly fluctuating E_d value. It must be remembered that HYDROLIGHT's predictions correspond to measurements averaged over times that are long with respect to any gravity wave, but that are short with respect to the time for the sun's position to change by more than a few degrees.

5.4 Comments on the Mathematical Algorithms

Finally, it should be noted that some of the numerical algorithms used in HYDROLIGHT 3.0 have, on rare and pathological occasions, failed in one way or another. Some of the *Numerical Recipes* mathematical algorithms have been criticized as being obsolete or worse [see, for example, Shampine (1987)]. Nevertheless, the *Numerical Recipes* algorithms are used in HYDROLIGHT 3.0 because they are inexpensive, widely available, simple to use, and they provide satisfactory results in all but extreme cases.

One notable situation in which a *Numerical Recipes* algorithm sometimes fails arises if the depth profile of an IOP (such as a, b, or an internal source strength representing bioluminescence) is discontinuous with depth, that is, if the value of the IOP jumps from one value to another in a stairstep fashion. In this case, the *Numerical Recipes* Runge-Kutta algorithm odeint may "hang up" in trying to get past the discontinuity: the algorithm

keeps trying ever smaller steps Δz in order to get past the discontinuity and still satisfy its naive error criterion for the accuracy of the solution. This problem can be avoided by slightly smoothing the discontinuity in the IOP, so that the jump in the IOP value occurs over a small but nonzero depth interval. An example of such smoothing is seen in the distributed version of routine s0biolum, which models an internal layer of bioluminescing micro-organisms. Another example of smoothing of a discontinuous depth profile of IOP's was seen in Example 3 of Section 4.4.3, which used a cubic-spline fit to measured data. In addition to serving the purpose of interpolating between the measured values, the cubic-spline depth profile of the IOP's was sufficiently smooth that no numerical problems arose. It is unlikely that any physically realistic situation will encounter IOP depth profiles that are more extreme than the ones used in that example.

We have already commented that running HYDROLIGHT to extreme optical depths must sooner or later lead to problems with the computer representation of the resulting small numbers. However, HYDROLIGHT on occasion has been run to 50 optical depths without encountering any numerical difficulties. Such depths are far beyond what is required in routine oceanographic applications and far beyond what could be reached with Monte Carlo models.

The invariant imbedding algorithms themselves appear to be very robust and never have caused any numerical difficulties, even for extreme cases such as $\omega_o = 0.01$ or 0.99.

5.5 Comments on Debugging

Previous versions of the HYDROLIGHT code have been extensively debugged using internal checks (such as conservation of energy), by simulation of special cases for which analytic solutions exist, by comparison with measured data (see *Light and Water*, Sections 11.2 and 11.3), and by comparison with independent Monte Carlo and discrete ordinates models [see Mobley et al. (1993)]. In all cases agreement has been excellent. However, version 3.0 of the code is extensively rewritten from the previous versions, and it is possible that errors may have been introduced during the rewriting. If so, these are likely to be found quickly by the first few users of version 3.0. Anyone encountering suspicious results should contact the author.

6. REFERENCES

Dongarra, J. J. and E. Grosse, 1987. Distribution of mathematical software via electronic mail, *Commun. ACM*, **30**(5), 403-407.

Gregg, W. W. and K. L. Carder, 1990. A simple spectral solar irradiance model for cloudless maritime atmospheres, *Limnol. Oceanogr.*, **35**(8), 1657-1675.

Harrison, A. W. and C. A. Coombes, 1988. An opaque cloud cover model of sky short wavelength radiance, *Solar Energy*, **41**(4), 387-392.

IDL® (Interactive Display Language) is a product of Research Systems, Inc., Boulder, CO 80303. Send fax enquiries to 303-786-9909 or email to support@rsinc.com.

Kasten, F. and G. Czeplak, 1980. Solar and terrestrial radiation dependent on the amount and type of cloud, *Solar Energy*, **24**, 177-189.

Kirk, J. T. O., 1994. *Light and Photosynthesis in Aquatic Ecosystems*, Cambridge University Press, Cambridge, 509 pages.

Kneizys, F. X., E. P. Shettle, L. W. Abreu, J. H. Chetwynd, G. P. Anderson, W. O. Gallery, J. E. A. Selby, and S. A. Clough, 1988. Users guide to LOWTRAN 7, Air Force Geophysics Lab, Rept AF6L-TR-88-0177, Hanscom AFB, MA 137 pages.

LOWTRAN, see Kneizys, et al. (1988).

Maritorena, S., A. Morel, and B. Gentili, 1994. Diffuse reflectance of oceanic shallow waters: influence of the water depth and bottom albedo. *Limnol. Oceanogr.*, **39**(7), 1689-1703.

Mobley, C. D., 1988. A numerical model for the computation of radiance distributions in natural waters with wind-roughened surfaces, Part II: Users' guide and code listing, Tech. Memo. ERL PMEL-81, NOAA Pacific Marine Environmental Lab, Seattle, 170 pages.

Mobley, C. D., 1994. *Light and Water: Radiative Transfer in Natural Waters*, Academic Press, San Diego, 592 pages.

Mobley, C. D., 1995. *The Optical Properties of Water*, Chapter 43 in *Handbook of Optics, Second Edition, Volume I*, M. Bass, Editor in Chief, McGraw-Hill and Optical Society of America, New York, 56 pages.

Mobley, C. D. and R. W. Preisendorfer, 1988. A numerical model for the computation of radiance distributions in natural waters with wind-roughened surfaces. Tech. Memo. ERL PMEL-75, NOAA Pacific Marine Environmental Lab, Seattle, 195 pages.

Mobley, C. D., B. Gentili, H. R. Gordon, Z. Jin, G. W. Kattawar, A. Morel, P. Reinersman, K. Stamnes, and R. H. Stavn, 1993. Comparison of numerical models for computing underwater light fields, *Appl. Opt.*, **32**, 7484-7504.

Preisendorfer, R. W. and C. D. Mobley, 1985. Unpolarized irradiance reflectances and glitter patterns of random capillary waves on lakes and seas, by Monte Carlo simulation. Tech. Memo. ERL PMEL-63, NOAA Pacific Marine Environmental Lab, Seattle, 141 pages.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. *Numerical Recipes in FORTRAN: The Art of Scientific Computing, Second Edition*, Cambridge University Press, Cambridge, 963 pages. The associated FORTRAN code and licenses can be purchased from Numerical Recipes Software, Cambridge, MA 02238. Send fax enquiries to 617-863-1739 or email inquiries to orders@nr.com.

Shampine, L. F., 1987. Review of *Numerical Recipes: The Art of Scientific Computing*, *Amer. Math. Monthly*, **94**(9), 889-892.

Spinrad, R. W., K. L. Carder, and M. J. Perry, 1994. *Ocean Optics*, Oxford University Press, New York, 283 pages.